REPORT NO. GDC DDF66-007

AUTOMATIC MALFUNCTION ANALYSIS BY DISCRETE NETWORK SIMULATION

GPO PRICE $ _____

CFSTI PRICE(S) $ _____

Hard copy (HC) 3.00

Microfiche (MF) 1.30

ff 653 July 65

GENERAL DYNAMICS
Convair Division

REPORT NO. GDC DDF66-007

AUTOMATIC MALFUNCTION ANALYSIS BY DISCRETE NETWORK SIMULATION

OCTOBER 1966

Submitted to
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
HUNTSVILLE, ALABAMA
under
CONTRACT NAS8 – 20016

Prepared by
CONVAIR DIVISION OF GENERAL DYNAMICS
HUNTSVILLE, ALABAMA

TABLE OF CONTENTS

ILLUSTRATIONS

TABLES

SUMMARY

Automatic Malfunction Analysis data has been generated using Discrete Network Simulation for selected networks of the Saturn 1C Stage. Discrete Network Simulation (DNS) uses a family of computer programs developed by General Dynamics Convair to simulate the real time operation of a discrete system. DNS techniques and their application are described in the Phase I Final Report of this contract. (Reference 1-3).

A brief description of the three (3) original DNS programs is included in this report, since the information they generate provides the data for the Automatic Malfunction Analysis (AMA). The instructions for building a DNS model are included in this volume as Appendix A.

Automatic Malfunction Analysis is an analytical technique and a computer program that determines the cause of an indicated "No-Go" on any discrete signal being monitored. The AMA Editor Program condenses and formats the AMA data so it may be stored on tape and retrieved by the checkout computer.

A logic model, system simulation, and AMA data were created for both the Engine Cutoff Networks and the Power and Control System of the Saturn 1C. Specific examples of AMA data for the Engine Cutoff Networks as they were retrieved by the checkout computer have been documented. The data format for the two tapes generated by the AMA Editor indicates the checkout computer's use of the data. The Programmers Reference Manuals are included as Appendixes.

1. Discrete Network Simulation Programs, Appendix B.

2. Automatic Malufnction Analysis Programs, Appendix C.

1/DISCRETE NETWORK SIMULATION

The prime purpose of Discrete Network Simulation methodology is to provide a set of analytical tools capable of conducting thorough, accurate and rapid analysis of complex systems. The methodology consists basically of:

A System network model.

A set of three computer programs which operate and activate the model.

These programs provide a realistic analysis and prediction of system performance before or after the hardware system is constructed. It is another form of testing; the results are as valid as those obtained by the more common hardware test procedures.

The model, Boolean equations, is the input to the Down Translation and Culling Program which condenses and formats the engineering data for processing in the computer. The Preprocessor Editor Program analyzes the equations to establish the interrelationships of the model used in both the Simulation and AMA programs.

The Discrete Network Simulator (DNS) chronologically simulates events occurring due to the interactions among elements in a system network. Each "event," a Boolean change of state, is the result of a logical cause and effect relationship among elements in the system. The system modeled for the simulation may be a switching circuit, man/machine interaction, or any network where the component or subcomponent interrelations may be defined logically.

1.1 DOWN TRANSLATION AND CULLING PROGRAM

The amount of memory in a computer places a size limitation on the amount of data that can be processed at one time. A detailed simulation of a complex system at component level utilizes a relatively large percentage of this "memory" to store the model. The Down Translating and Culling Program (DT&C) was developed to augment the DNS programs and provide a more efficient utilization of computer "memory."

In the first DNS computer programs each variable or component in the equation had to be described with a coded symbol made up of less than six characters. This required a secondary coding of each component in a more condensed form than

would normally appear on a schematic diagram, making the verification and check-out of the simulation difficult. The DT&C Program permits the use of descriptive variable names of up to twenty-four characters. Variable names are constructed from the actual name and location of a component within a system. Therefore, the logic equations are written in descriptive engineering terms that are readily identifiable, even to those unfamiliar with the program.

Application of this program to the simulation of the S-1C Engine Cutoff System permits the variables to be described as they appear on the Schematic Drawings. The logic equations are written identifying every variable in each system. Each serial path is a detailed description of the circuitry including cable pins and plugs, diodes, fuses and power distribution buses. When the Down Translation Program processes the system model, it assigns to each variable an arbitrary three-letter code. This three-letter symbol is subsequently used in the remaining DNS programs. When a printout of the processed data is obtained the three-letter code is replaced by the original engineering term or descriptive word simplifying the analysis of the processed data.

The following outputs are generated by the DT&C Program:

1. A tape containing the coded equations and the time parameter data. This tape is an input to the subsequent Preprocessor Editor. Information on this tape is in Binary Coded Decimal (BCD) format.

2. A binary tape that includes a table of the complete variable names in their original format. This tape is required as an input to the AMA Program.

3. A print out of the time parameters and equations, exactly as they appear on the input card deck.

Figure 1 shows a page from the DT&C printout of the logic model for the Engine Cutoff System. This portion of the printout shows the variables and their time parameters exactaly as key punched during the model preparation phase. The only additional information on the printout is the three-letter variable code names arbitrarily assigned to each of the engineering designations for the variables.

All components on the drawing are written in the equations for the sake of completeness and to include the ability to investigate the effect of their failure during subsequent analysis. To cut down the size of the "model" for the actual simulation only those variables that are active (change state as a result of normal system actions) are retained for the actual simulation. Variables such as pins, fuses, diodes, etc., do not change state as a result of "system reactions", but do change state if they fail. These variables are classified as "Inactive" and are

DISCRETE NETWORK SIMULATOR-DONNTRANSLATED TIME CARDS

C3W	CØNT115A3A7<18C46	S	4S	5S	6S	4S	5S	6ADUM	CO
C3X	CØNT115A3A7<28C13	S	4S	5S	6S	4S	5S	6ADUM	CO
C3Y	CØNT115A3A7<28C45	S	4S	5S	6S	4S	5S	6ADUM	C1
C3Z	CØNT115A3A7<28C46	S	4S	5S	6S	4S	5S	6ADUM	CO
C3A	CØNT115A3A7<38C13	S	4S	5S	6S	4S	5S	6ADUM	CO
C3B	CØNT115A3A7<38C45	S	4S	5S	6S	4S	5S	6ADUM	C1
C3C	CØNT115A3A7<38C46	S	4S	5S	6S	4S	5S	6ADUM	CO
C3D	CØNT115A3A7<41C46	S	4S	5S	6S	4S	5S	6ADUM	CO
C3E	CØNT115A3A7<51C13	S	4S	5S	6S	4S	5S	6ADUM	CO

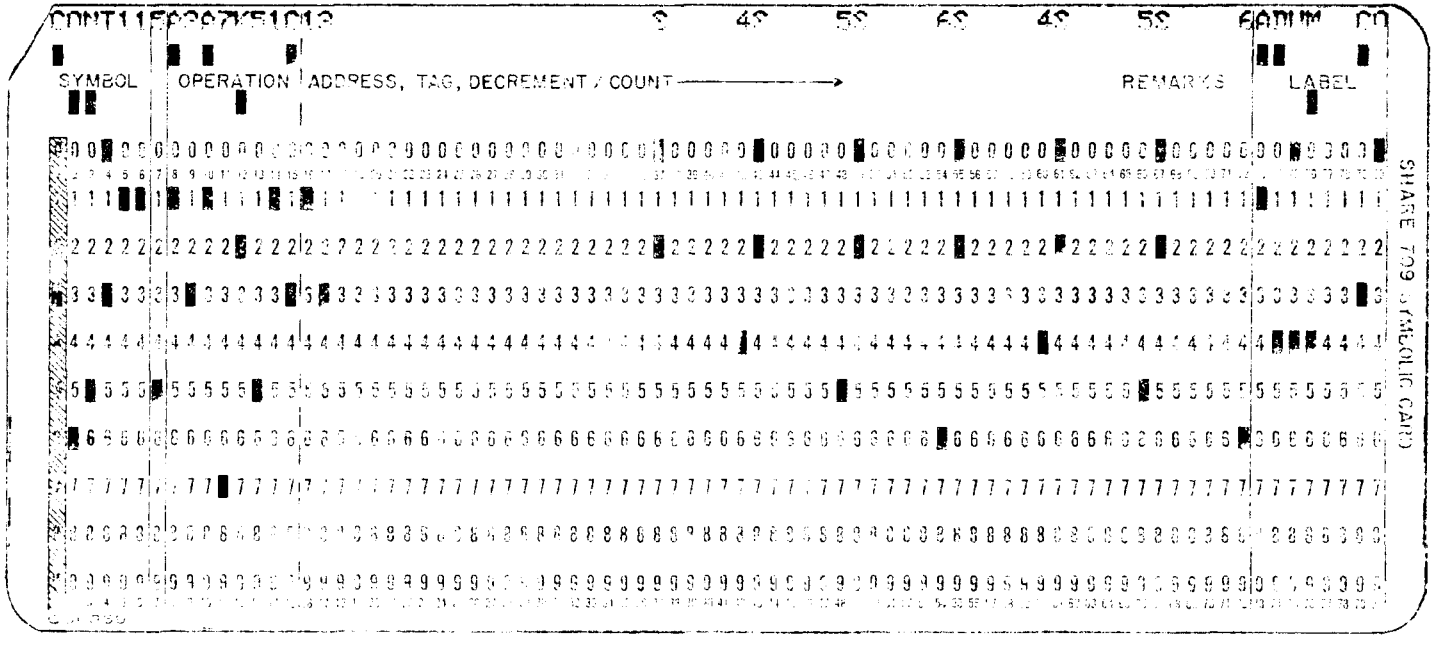


Figure 1. Time Card and Printout From DT&C

culled from the logic equations by the DT&C Program. Eliminating these variables reduces the computer running time for the simulation program.

Figure 2 shows a portion of the DT&C printout that illustrates the culling sequence. One asterisk indicates the original equation as it was punched on the card. Two asterisks denote the culled equation; all variables that were classified as Inactive on the time-card are removed from the equation. Three asterisks designate the translated equation (the culled equation with the engineering designation replaced by the three-letter code).

Even though the inactive variables are not used for a "normal" simulation run, they cannot be overlooked as failure possibilities. The DT&C Program was modified to assign a three-letter code to the inactive variables. This information is included on the binary output tape but is not on the printed output.

A second modification was made to permit a choice of specific types of variables to be considered as failure candidates by the AMA Program by inserting two cards in the DT&C data deck. These provide a listing of the type variables to be processed for failure analysis. The variable identification is taken from the "variable type coding" in column 79 of each time parameter card.

The following additional refinements were incorporated in the program to facilitate verification of the logic equations:

1. Each equation is arbitrarily assigned a reference index number that appears on the extreme right side of the equation format printout as shown in Figure 2.

2. All the variables that are classified as "Inactive" with the variable classification in columns 73 to 80 of the time parameter card are listed. Figure 3 shows a portion of the S-1C Cutoff System printout that illustrates the format.

3. Variables that do not contain a time parameter card are classified as Unspecified as shown in Figure 4. This is an example of a DT&C printout generated for model validation only. The number on the left is the equation number in which the variable is located. The variable will be listed each time it appears in an equation. As previously stated, the DT&C Program culls out Inactive Variables; it also culls out variables that do not contain a time parameter card. If the "culled" equation does not have an Active variable on the left of the equal sign, the entire equation is deleted. It is also deleted if the variable on the left is active, but there are no active variables to the right of the equal sign. In either case, the message "this equation has been deleted" is printed as illustrated in Figure 4.

```
                    DISCRETE NETWORK SIMULATOR - DECNTRANSLATION OUTPUT
        *-ORIGINAL EQUATION  **-CULLED EQUATION ***-TRANSLATED AND CULLED EQUATION
  *                    DT+C SIC ENG CUTOFF SYS  SH 34-52
  *   DT + C          GSE FOR ECO SH 34 TO 52
  *          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  *       DI64    = /PIN6A4J20E *NODE6A4J21B.                    615 46 1    1473
  **      DI64=NODE6A4J21B.                                      615 46 1    1473
  ***      AAD   = AUP   .                                       615 46 1    1473
  *       DI65    = /PIN6A4J20F *NODE6A4J21K.                    615 47 1    1474
  **      DI65=NODE6A4J21K.                                      615 47 1    1474
  ***      AAE   = AUQ   .                                       615 47 1    1474
  *       DI66    = /PIN6A4J20G *NODE6A4J21U.                    615 48 1    1475
  **      DI66=NODE6A4J21U.                                      615 48 1    1475
  ***      AAF   = AUT   .                                       615 48 1    1475
  *       NODE6A4J18N   = LEG3N6A4J18N.                          619  5 1    1774
  **      NODE6A4J18N=LEG3N6A4J18N.                              619  5 1    1774
  ***      BOE   = DMN   .                                       619  5 1    1774
  *       CONT12A3K1J8SPSQ  = COIL12A3K1J8PPSJ.                  615A50 1    1514
  **      CONT12A3K1J8SPSQ=COIL12A3K1J8PPSJ.                     615A50 1    1514
  ***      CVT   = AOM   .                                       615A50 1    1514
  *       CONT12A3K1J8SQSR  = COIL12A3K1J8PPSJ.                  615A51 1    1515
  **      CONT12A3K1J8SQSR=COIL12A3K1J8PPSJ.                     615A51 1    1515
  ***      CVU   = AOM   .                                       615A51 1    1515
  *       COIL5A5K615J38JK    = BUS5A5-21D1 * NODE5A5J28V.       621C41 1    2172
  **      COIL5A5K615J38JK=BUS5A5-21D1*NODE5A5J28V.              621C41 1    2172
  ***      BSI   = ASB  * BJP   .                                621C41 1    2172
  *       COIL5A5K616J38TU    = BUS5A5-21D1 * NODE5A5J28X.       621C42 1    2173
  **      COIL5A5K616J38TU=BUS5A5-21D1*NODE5A5J28X.              621C42 1    2173
  ***      BSJ   = ASB  * BJQ   .                                621C42 1    2173
  *       LEG3N5A5J40SG       = /CONT5A5K87J35LM *BUS5A521D11.   621F61 1    2268
  **      LEG3N5A5J40SG=/CONT5A5K87J35LM*BUS5A521D11O.           621F61 1    2268
  ***      DFU   =/ CIL  * BET   .                               621F61 1    2268
  *       LEG3N5A5J40SH = /CONT5A5K88J35YZ *BUS5A51D111.         621F19 1    2269
  **      LEG3N5A5J40SH=/CONT5A5K88J35YZ*BUS5A51D111.            621F19 1    2269
  ***      DFV   =/ CIM  * BFS   .                               621F19 1    2269
```

Figure 2. Model Equations From DT&C

INACTIVE VARIABLES

LEG1N6A4J22SW	IIUS	YO
LEG1N6A4J28A	IIUS	YO
LEG1N6A4J28J	IIUS	YO
LEG1N6A4J28SB	IIUS	YO
LEG1N6A4J28SJ	IIUS	YO
LEG1N6A4J28T	IIUS	YO
LEG1N7A3K5C1	IDRD	YO
LEG2N7A3K5C1	IDRD	YO
SØL51A13A4	IIFU	NO
SØL4C1A6A2	IIFU	NO
F401A6YF1	I	V1
NFCIC2UPL107	I	V1
NFCIC2UPL108	I	V1
V401A6VM1	I	V1
V401A6VM4	I	V1
DIØDE115A2A6CR1Ø	II I	DO
DIØDE115A2A6CR3Ø	II I	DO
DIØDE115A3A1CR1Ø	II I	DO
DIØDE115A3A1CR2Ø	II I	DO
DIØDE115A3A1CR3Ø	II I	DO
DIØDE115A3A1CR4Ø	II I	DO
DIØDE115A3A1CR5Ø	II I	DO
DIØDE115A3A1CR6Ø	II I	DO
DIØDE115A3A1CR7Ø	II I	DO
DIØDE115A3A1CR8Ø	II I	DO
DIØDE115A3A1CR9Ø	II I	DO
DIØDE115A3A4CR19Ø	II I	DO
DIØDE115A3A4CR2Ø2	II I	DO
DIØDE115A3A4CR21Ø	II I	DO
DIØDE115A3A4CR222	II I	DO
DIØDE115A3A4CR23Ø	II I	DO
DIØDE115A3A4CR41Ø	II I	DO
DIØDE115A3A4CR422	II I	DO

Figure 3. Inactive Variables from DT&C

UNSPECIFIED NAMES LIST

EQUATIØN	NAME
995	PIN115W43P1M
995	PIN115W43P5A
995	PIN115W43P5B
995	PIN115W43P1T
996	PIN115W43P1S
996	PIN115W43P6A
996	PIN115W43P6B
996	PIN115W43P1X
997	PIN115W43P1R
997	PIN115W43P7A
997	PIN115W43P7B
997	PIN115W43P1W
1123	LEG3N6A3J12SE
1125	NØDE6A3J12SH
1125	LEG3N6A3J12SH
1125	THIS EQUATIØN HAS BEEN DELETED
1126	LEG3N6A3J12SI
1131	PIN5A2J14B
1131	PIN5A2J39SI
1131	PIN6A3J39SI
1132	PIN5A2J14C
1132	PIN5A2J39SE
1132	PIN6A3J39SE
1133	PIN6A3J39SK
1133	PIN5A2J14H
1133	PIN5A5J8H
1134	PIN6A3J39SM
1134	PIN5A2J14J
1134	PIN5A5J8J
1135	PIN6A3J39SN
1135	PIN5A2J14K
1135	PIN5A5J8K
1136	PIN6A3J12SE

Figure 4. Unspecified Names from DT&C

1.2 THE DNS PREPROCESSOR EDITOR PROGRAM

The DNS Preprocessor Editor Program (PREP-ED) is a combined form of earlier Convair DNS Preprocessor & DNS Editor Programs. This program utilizes the culled and coded output of the DT&C Program and is required to "order" or "process" the data for the DNS simulation and AMA programs. The program provides a series of tables that define the interdependancy relationship of each active variable within a model. Self-checking diagnostic features are built into the program to insure that every variable on the left hand side of any equation also appears on the right hand side of an equation unless it is identified as a terminal. It checks that activation times have been supplied for each variable and produces a listing of the logic equations and timing information for reference.

Figure 5 is a composite sample of the three types of printout produced by the Preprocessor portion of the program. These printouts are mainly for reference use during the model building process and have no direct analytical value.

Several additional program options are available that enhance the versatility of the overall AMA Program and provide different print formats. These options are selected by using specified control cards in the data section of the program binary deck.

The "Index" control card produces an alphabetical list of all coded variables, the corresponding internal code numbers, and a classification of the variables.

Initiator - used only in the right hand side of an equation.
Transactor - used on left and right hand sides of different equations.
Terminal - used only on the left hand side of an equation.

This list is followed by a "Variable Reference" Table in which each variable name is printed out followed by a list of functions (Equations) in which it occurs. Figure 6 is a composite sample of the printout.

The "Index Full" Control Card creates two additional tables which printout following the two above. They are the "Terminal-Variable" table and the "Variable-Terminal" table. The Terminal-Variable table lists the terminals on the left hand side of the page and follows each terminal with a list of variables that are directly or indirectly affected by the subject terminal. The Variable-Terminal list prints the variable on the left hand side of the page followed by a listing of the Terminals on which that variable will have an affect. These tables are illustrated in Figure 7.

The 'Index Logic' Control Card prints only the Variable-Terminal and Terminal-Variable tables.

```
*TERMINALS                                                                    CCCC
*                                                                             CCCC
*INPUT AMA TAPE                                                               CCCC
*                                                                             CCCC
*TIMES
AAA          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAB          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAC          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAD          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAE          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAF          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAG          S    CS    CS    C      S    CS    CS    C                      AIFU  I
AAH          S    CS    CS    C      S    CS    CS    C                      AIFU  I

*EQUATIONS
*                        DT+C S1C ENG CUTOFF SYS  SH 34-52
*  DT + C          GSE FOR EC0 SH 34 TO 52
*          * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 AFI  = BBD  .                                                    35  79 1
 AFH  = BBC  .                                                    35  14 1
 AFY  = CYS  .                                                    35   1 1
 BHE  = CWI  + CWA  .                                             35   2 1
 AFW  = CXS  .                                                    35 119 1
 AFX  = CXT  .                                                    35 120 1
 AGH  = CZJ  + CZK  + CZL  + CZM  + CZN  + CZ0  .                 35   9 1
 AGM  = DAC  .                                                    35  13 1
 BBC  = ANR  */ BDH  .                                            35/621E

 *NAMES
 AAA          CI8
 AAB          CI11
 AAC          CI15
 AAD          CI64
 AAE          CI65
 AAF          CI66
 AAG          CI67
 AAH          CI68
```

Figure 5. Preprocessor Editor Composite Printout

KEY	ID	VARIABLES			
AAA	1	TERMINAL			
AAB	2	TERMINAL			
AAC	3	TERMINAL			
AAD	4	TERMINAL			
AAE	5	TERMINAL			
DØZ	2418	TRANSACTØR			
DPA	2419	TRANSACTØR			
DPB	2420	TRANSACTØR			
DPC	2421	TRANSACTØR			
DPD	2422	TRANSACTØR			
DPE	2423	TRANSACTØR			
DPF	2424	TRANSACTØR			
DPG	2425	TRANSACTØR			
INITIATØR	170	TRANSACTØR	2125	TERMINAL	130

VARIABLE	FUNCTIØNS									
468 ARZ	CEK	,CEJ	,CEI	,DØY	.					
469 ASA	AØF	,AØG	,AØH	,AØI	,AØJ	,AØK	,AØL	,BRN	,BRØ	.
470 ASB	AØA	,AØB	,AØC	,AØD	,AØE	,BRQ	,BRR	,BRS	,BRT	,BRU ,
	BRX	,BRY	,BRZ	,BSA	,BSB	,BSC	,BSD	,BSE	,BSF	,BSG ,
	BSJ	,BSK	,BSL	,BSM	,BSN	,BSØ	,BSP	,BRP	.	
471 ASC	CCC	,CHE	,CHF	,CHG	,CGJ	,CGI	,DFP	,DFQ	,DFR	,DFS ,
	CFV	,CFW	,CFX	,DFØ	,CFY	,CFZ	,DGA	,DGB	,DGC	,CDJ ,
	CHB	,CHD	,CHH	,DHJ	,CHK	,CHØ	,DGZ	,CHA	,CHC	,CHN .
472 ASD	CCW	,DGH	,CHM	.						

Figure 6. Variable Classification and Function Table

TERMINAL REFERENCE TABLE

TERMINALS	VARIABLES
128 AEX	
129 AEY	
130 AEZ	
131 AFA	ABC ,ACR .
132 AFB	ABD ,ACR .
133 AFC	ABE ,ACR .
134 AFD	AAA ,AAB ,AAC ,AAV ,AAW ,AAY ,ABF ,ABI ,ABJ ,ABK ,
	ABN ,ABØ ,ABP ,ABQ ,ABR ,ABS ,ABT ,ABU ,ACB ,ACD ,
	ACG ,ACH ,ACI ,ACJ ,ACK ,ACL ,ACM ,ACQ ,ACS ,ACV .
	ACY ,ACA ,ACQ ,ACR ,ACS ,ACT .

TERMINAL REFERENCE TABLE

VARIABLES	TERMINALS
120 AEP	AHS ,AIB ,AIY ,AKW ,AMW ,AØG ,AQJ ,AQV ,ARØ ,ARR ,
	ASC ,AZA ,AZQ ,AZR ,BBQ ,BBR ,BEF ,BJZ ,BLW ,BMW ,
	BZN ,BZQ ,BZU ,CDM ,CDP ,CDT ,CFY ,CKA ,CQY ,CTZ ,
	DCL ,DFT ,DGQ ,DJU ,DKX .
121 AEQ	AIC ,AIG ,AJI ,AKX ,ANH ,AØF ,AQP ,ARP ,ARR ,ASA ,
	AZM ,BAM ,BAN ,BCK ,BCL ,BEG ,BKE ,BLX ,BMX ,BRX ,
	BZZ ,CAD ,CAE ,CDZ ,CED ,CEE ,CFZ ,CKB ,CKU ,CTG .
	DCQ ,DFU ,DGV ,DJV ,DKY .

Figure 7. Variable Reference Tables

Discrete Inputs to the RCA110 Computer as well as lights, analogue discretes, and digital event evaluator (DEE) signals, were among the variables that the program could classify as terminals. For this application only, Discrete Inputs are classified as terminals, as the DI's are the only reference points for malfunction analysis available to the checkout computers. To accomplish this, a "Terminal" control card was added, which instructs the Preprocessor Editor Program to recognize as terminals only those variables whose time parameter cards contained an "I" in Column 79.

The variable terminal and terminal variable tables will reflect those variables so designated.

The combined program produces a second binary tape that contains all the necessary variable dependance information for the AMA Program, including a table with each designated "Terminal" and the variables that can affect it in any way.

1.3 DISCRETE NETWORK SIMULATION PROGRAM

After the system model has been processed through the DT&C and PREP-ED Programs, system operation can be simulated using the DNS Program. The model, as written, represents the system in a static condition. A set of drive functions (Input Deck), is then required to establish the model in the initial condition for the simulation and to represent the activities to be simulated. The Input Deck contains the state of each Input required to "drive" the model and the time that the Input will be processed. The results of the simulation are recorded on the output tape. The printout includes:

a) The order of events occurring
b) The time of each occurrence
c) A list showing the state of all variables at any selected time.

The program first establishes an initial condition for the state of the variables in the model. On the basis of this state, it examines all equations in the model and the logic predicts what variables will change state. The simulation represents real time: therefore, after the prediction has been made, the program looks up the activation times for the variables changing state, and imposes that time delay on the program before allowing the predicted changes to occur. This process is indicated on the printout, Figure 8. "Input" is the code word indicating that the state of this variable is being set by an external command in the input deck. The code word, "Enter" on Figure 8. indicates that this variable is changing state due to the logic of the equations combined with the computer program. On this figure there are several variables listed on the left side which have no code word. The absence of the code word indicates that these are predicted activities that may occur as a result of activity described immediately above.

TYPE	DESCRIPTION	VALUE	NO. OF EVENTS	HOURS	MINUTES	SECONDS
INPUT	D0771	1	5			.400
	LEG2N6A3J6P	1				.400
	D0771	0				.430
ENTER	LEG2N6A3J6P	1	6			.400
	N0DE6A3J6P	1				.400
ENTER	N0DE5A3J6P	1	6			.400
	P120W3P2P	1				.400
ENTER	P120W3P2P	1	6			.400
	J120W3P2P	1				.400
ENTER	J120W3P2P	1	6			.400
	C0IL115A5K5SET	1				.405
ENTER	C0IL115A5K5SET	1	6			.405
	C0IL115A5K5RESET	0				.405
	J120W3P2SE	1				.405
ENTER	C0IL115A5K5RESET	0	7			.405
ENTER	J120W3P2SE	1	6			.405
	P120W3P2SE	1				.405
ENTER	P120W3P2SE	1	6			.405
	LEG3N5A2J35SS	1				.405
ENTER	LEG3N5A2J35SS	1	6			.405
	N0DE5A2J35SS	1				.405
ENTER	N0DE5A2J35SS	1	6			.405
	C0IL5A2K151J26PPSS	1				.405
	DI1207	1				.405
ENTER	C0IL5A2K151J26PPSS	1	7			.405
	C0NT5A2K151J26STSJ	1				.410
ENTER	DI1207	1	7			.405
ENTER	C0NT5A2K151J26STSJ	1	6			.410
	LITE388A6DS4G	0				.410
ENTER	LITE388A6DS4G	0	6			.410
ENTER	D0771	0	5			.430

Figure 8. Simulation Printout

This simulation represents real time. The time of the simulation is printed in the columns on the right hand side of the sheet, and as indicated by the headings, can be described in days, hours, minutes, or seconds, The seconds are resolved down to the nearest millisecond. The time printed for each activity line represents the actual time for the activity to occur. The time listed for the predicted event is the time that the event should occur based upon the time required for that component to activate.

In the center of the page there is a column headed "Number of Events." This is a bookkeeping function for the Simulator Program which lists the number of predicted events that have to be satisfied on the basis of either the imputs or actions that have already taken place. At the end of a simulation activity, this column will be reduced to zero.

The final printout from the Simulation is a representative state list. If a "List" control card is included as the last card in the Input Deck, the program will summarize the results of the simulation by printing a list giving the state (0 or 1) of all the variables in the model. If a second control card, "List 1" is included, only those variables that have a "1" state are listed as shown in Figure 9.

For the SIC Checkout, state lists are created during the simulation process by the insertion of a "List" control card for each "Scan" in the test procedure. The control cards will cause a state list to be printed at the completion of each scan.

Time and event relationships between the simulation run and the actual test being conducted on the Saturn 1C were accomplished by modifying the program to add the test block, step and sub-step identification to the Simulation "List" Control Cards. Lists of variable states generated by the Simulation Program are thereby indexed to a particular time and scan.

See Appendix B Users Reference Manual for an explanation and proper use of the control cards for the Simulation Program.

```
*TRANSLATIØN MØDE
*LIST 1

          THE S1C ENGINE CUTØFF SYSTEM DTC1392 PREP3395

 *LIST  AT          146000    NØ.      1

      14  DI266                                =          1.
      16  DI268                                =          1.
      18  DI270                                =          1.
      20  DI272                                =          1.
      22  DI274                                =          1.
      24  DI276                                =          1.
      78  NØDE5A2J2S                           =          1.
      80  NØDE5A2J2U                           =          1.
      88  NØDE5A2K127J22X                      =          1.
      90  NØDE5A5J12BB                         =          1.
      93  NØDE5A5J12GG                         =          1.
      96  NØDE5A5J12LL                         =          1.
     260  CØIL5A7K426J5AB                      =          1.
     261  CØIL5A7K426J10JK                     =          1.
     262  CØIL5A7K427J7JK                      =          1.
     263  CØIL5A7K427J10TU                     =          1.
     264  CØIL5A7K434J18TU                     =          1.
     265  CØIL5A7K434J21JK                     =          1.
     639  CØNT5A2K163J29WX                     =          1.
     640  CØNT5A2K17CJ36SESF                   =          1.
     645  CØNT5A7K426J10RS                     =          1.
     646  CØNT5A7K427J10SAZ                    =          1.
     647  CØNT5A7K434J18SAZ                    =          1.
     648  CØNT5A7K435J18SHSI                   =          1.
     861  LEG3N6A4J34U                         =          1.
     878  LEG2N6A5J24SX                        =          1.
*END LIST
```

Figure 9. Simulation State List

2/ AUTOMATIC MALFUNCTION ANALYSIS

2.1 PRINCIPLE OF AUTOMATIC MALFUNCTION ANALYSIS

Automatic Malfunction Analysis is an analytical technique that determines, independently of the actual hardware, the cause of an indicated 'No-Go' on any of the discrete signals being monitored. For each available discrete indicator, AMA has determined before the test the possible single failures that could cause the 'No-Go'. The DNS Program defines the normal status of all components in a complex discrete network at any instant in time. The status of the system, or the state of the variables, is the input to a variation of the DNS Program. One variable can be analyzed to determine what changes in the system's state would cause that variable to be in the opposite state. The result is a list of variables whose failure (opposite from normal) would cause the variable being investigated to be in the wrong condition.

Specifically, the model represents the electrical networks that make-up the Engine Cutoff System for the S-1C. The normal condition represents the model reacting to the stimulus provided by the checkout procedure for the Engine Cutoff System. The results of the analysis are lists of components that could cause one of the indicators to indicate a 'No-Go' condition.

Figure 10 shows the methodology of Automatic Malfunction Analysis. Given the equation and the values for the elements of the equation, it can be seen that either 'A' or 'B' being '0' would cause the 'DI' to be '0' or off. Thus, 'A' and 'B' are both possible failure candidates, In the equation for element 'A', the term (E*F) already has a value of '0'; so that the failure of element 'E' would not cause a change of state in 'A'. Thus, both 'C' and 'D' are possible causes of failure for the 'DI' being investigated, Since 'C' is an initiator, (is not further defined) it is a failure candidate affecting the 'DI'. However, 'D' is defined by the equation (D=R+S). Since both 'R' and 'S' have a value of '1', a single failure of either would not cause 'D' to become '0'. Therefore, neither elements 'D', 'R' nor 'S' are failure candidates. 'B' is defined by the equation, (B=X+Y+Z). Both 'Y' and 'Z' are '0', the element 'X' makes 'B' equal to '1'. In the absence of additional terms defining 'X', it becomes a failure candidate for the 'DI'.

	POSSIBLE CAUSE	FAILURE CANDIDATE
GIVEN		
DI = A * B. 1 1 1	A, B	
WHAT SINGLE CHANGE (FAILURE) WILL CAUSE DI TO BECOME ZERO.		
A = (C*D) + (E*F). 1 1 1 1 0	C, D	C
B = X+Y+Z 1 1 0 0	X	X
D = R + S 1 1 1		

* = AND
\+ = OR

Figure 10. Methodology of Automatic Malfunction Analysis

2.2 AUTOMATIC MALFUNCTION ANALYSIS PROGRAM

The complete analytical process described above is accomplished by the Automatic Malfunction Analysis Program. The inputs to this system from the Down Translation and Culling Program are the 'type' and 'classification' of the variables in the model, including which variables are DI's to be used as the basis for the analysis. The tables which define the inter-dependencies of the variables in the model are provided by the Preprocessor Editor Program. The state lists which define the status of the variables are generated by the simulation program. Each state list is the basis for an independent analysis at the point where there is a scan operator in the system's test procedure and is identified by block, step and sub-step. The output of the AMA Program, recorded on tape for additional processing, lists for each indicator (DI) the variables that could cause the DI to be in error at a particular block, step and sub-step.

2.2.1 AMA PROCESSES - To accomplish AMA there are three types of analysis required.

2.2.1.1 Zeros Analysis - This determines what circuit components can fail causing a terminal (DI) that is normally de-energized to be energized. For this type of analysis there are three criteria which must be satisfied; a path of connectivity exists from the component to the terminal, there is power to energize the path and it is possible for the component to fail.

2.2.1.2 Ones Analysis - This determines what circiut components can fail causing a DI that is normally energized to be de-energized.

2.2.1.3 Conditional Analysis - This is a special case of zeros and ones analysis required by the inter-dependency of the variables. Kirchoff's Leg Node technique is used for circuit modeling resulting in Leg Equations and Node Equations. Leg Equations are serial paths. If Zeros analysis is conducted for a Leg, and more than one component must fail to erroneously energize the Leg, the analysis is normally terminated. However, if the components are dependent upon one another or other variables not in the Leg, conditional analysis is required. In Figure 10 the equation D=R+S would require conditional analysis, as there is an inter-dependency between R & S. The analysis is performed inter-dependently for both R and S and the results of the two are compared. Only those candidates which are common to both the R and S analysis would be considered valid failure candidates.

2.2.2 FAILURE CANDIDATES - The type of failure candidates for each class of variable in the model must be specified for the AMA Program. Components can be failure candidates for 'ones', 'zeros' or both types of analysis. Components such as fuses are failure candidates for only 'ones' analysis, because their failure in the unenergized state cannot affect the indication of a zero value DI. The same is true for Legs connecting nodes in ones analysis. Diodes, coils and contacts are failure candidates for both ones and zeros analysis. Each type of component variable must

be analyzed to determine if it is a failure candidate and in which category it belongs.

2.2.3 COMMON POINTS - To maintain the efficiency of the AMA analysis it is essential to avoid redundant work. The detection and ranking of common points achieves this efficiency. Common points are model variables representing circuit components that are common to DI's (terminals being analyzed). The magnitude of the maximum number of DI's that can be affected by a component failure is called the common point level. Another important classification is that of common branch points. Common branch points are variables representing components which fan-out or upon which other portions of the discrete circuit depend. Examples of common branch points are:

1. Buses and grounds that affect different Legs
2. Nodes and terminals that affect different Legs
3. Coils that effect Legs in different circuits.

Common Branch Points coupled with their level or magnitude of possible terminal connectivity establish the mechanism for controlling the systematic methodical analysis of AMA. The general rule is to start with a defined DI and conduct analysis on all encountered circuit components at the same common point level. Components encountered at higher common point levels are noted as encountered and are to be processed later. These encountered components of higher common point levels are Common Branch Points that can effect DI's in addition to those at the present common point level. It is necessary to process encountered to-be-processed variables in ascending order; that is, finish common point level one, for Zeros and Ones, then analyze level two, etc., leaving conditional analysis equations to be processed last.

The DNS-Preprocessor produces a reference output showing for each model variable the corresponding defined terminals this component variable can possibly affect. From the previous discussion it is evident that the number of terminals indicated represents the Common Point Level.

2.3 THE AMA EDITOR PROGRAM

The AMA data generated must be immediately available during test operations. If a 'No-Go' occurs on a discrete input the checkout computer stops. The test conductor can request Automatic Malfunction Analysis data for the DI indicating 'No-Go'. The checkout computer will search the AMA data which has been recorded on tape and display to the test conductor those components that could cause that DI to be in error at that block, step and sub-step of the procedure. The AMA Editor Program puts the data in the proper format to be retrieved from tape on the checkout computer. The inputs to the AMA Editor Program are the AMA data and the dictionary of coded variables verses the engineering names from the Down Translation and Culling Program. There are two tapes produced by the Editor Program. The AMA tape contains the search keys for finding the desired information based on the step number and the DI in the failure. The malfunction set tape contains lists of

components that could cause the specified DI to be 'No-G0'. The malfunction set tape eliminates many redundant sets of component variables generated by the duplicate analysis of the same DI's for different steps and sub-steps in the procedure.

Figure 11 shows the format of the AMA tape produced by the AMA Editor Program. The first record contains 16 words of identification to be supplied by the using group. The second record contains 130 words and is used for tape identification. Words 1 through 4 show the test procedure and block number. The 63 words occupying position 5 through 67 of this record contain a DI profile similar to the profile in the RCA 110A computer memory. In this profile each bit of each word represents one DI-in all 1512 DI's can be represented. The profile identifies the DI's active in the specified test procedure for which AMA data is available. The second group of sixty-three words, 68-130 contains antoher DI profile identifying all other DI's for which AMA data is available on the malfunction set tape.

There are many systems which are activated for a particular test procedure, but the systems themselves remain essentially static during the procedure. An example would be the power and control system for the stage and GSE. The DI's associated with these systems maintain a fixed or static profile during the entire test run. In these cases a single AMA analysis for these DI's would be valid for the complete procedure. The second DI profile indicates which DI's have static malfunction information incorporated in the malfunction set tape.

The third record starts the AMA data. Words 1 through 4 contain block, step and sub-step numbers. Words 5 through 67 consist of a DI profile showing the normal 'on' or 'off' state of the DI's at this sub-step of the procedure. Word 68 indicates the number of DI's for which there is malfunction information in this sub-step. The remaining words from 69 up to the maximum of 255 consist of 2 word sets. The number of the DI and a malfunction set number for that DI for this sub-step of the procedure. (There is a maximum limit of 255 words in each record on this tape to maintain compatibility with the RCA 110A computer.)

The fourth record is either a continuation of the above information or it begins the AMA data for multiple independent single level malfunction profiles. Where a single malfunction would cause more than one DI to be 'No-Go' the AMA data must be keyed to a profile instead of a specific DI number. This is the function of this record. The second word of this record indicates the number of single level malfunction profiles contained in the record. Words 3 through 65 contain the profile with the individual bits set to '1' for those DI's in the 'No-Go' condition. Immediately following the profile is the malfunction set number for that set of DI's. Instead of recording a complete new 63 word profile, the original profile is up-graded to a new profile with a new malfucntion set. This new profile is then compared again with the actual profile in the RCA 110A computer. Word 67 of this record indicates the number of computer words to be changed to create a new profile. Word 68 is the number of the word to be changed and 69 is the replacement word itself. This 2 word pattern continues according to the number of groups indicated in word 67. When the new

AMA EDITOR PROGRAM

AMA Tape Format

FIRST RECORD:

RCA 110 Identification, 16 Words

SECOND RECORD:

(AMA Tape Identification, 130 Words)

Word 1:	TEST	(BCD Code)
Word 2:	Test Procedure Number	(BCD Code)
Word 3:	BLØC	(BCD Code)
Word 4:	Block Number	(BCD Code)
Words 5-67:	63 word identifying the DI's for which AMA data is available on this tape.	
Words 68-130:	63 word DI profile identifying all other DI's for which AMA data is available on the Malfunction Set Tape.	

THIRD RECORD:

Word 1:	BLØC	(BCD Code)
Word 2:	Block Number	(BCD Code)
Word 3:	Step Number	(BCD Code)
Word 4:	Sub-Step Number	(BCD Code)
Word 5-67:	DI Profile showing normal on/off state of DI at this sub-step.	
Word 68:	Number of DI's for which there is malfunction information in this sub-step.	
Word 69:	DI Number	
Word 70:	Malfunction Set Number	
Word 71 & on:	Repeat of two word groups (DI and Set Number)	

Maximum Record: 255 Words

FOURTH RECORD:

Either...	
Word 1	CONT
Word 2	Continuation of 2 word set
Word 3	

Figure 11. AMA Tape Format

Or...

Word 1:	MULT
Word 2:	Number of multiple independent single level malfunction profiles.
Word 3-65:	DI Profile with bits set to one for those DI's in 'no go' condition
Word 66:	Malfunction set number for the above profile
Word 67:	Number of words to be changed to create new profile
Word 68:	Number of Word to be changed
Word 69:	Replacement Word
	Continuation of two word groups equal to value of word 67
Next Word:	Malfunction set number for new profile
Next Word:	Repeat of above starting at Word 67 for as many profiles as indicated in word 2 of this record.

NEXT RECORD:

Either...

Word 1	CONT
Word 2	Continuation of previous record

Or...

Repeat of information starting at Third Record for new sub-step.

The tape is terminated with a termination-record followed by a double end-of-file.

Word 1:	ENDB	(BCD Code)
Word 2:	Block Number	(BCD Code)
Word 3:	TEST	(BCD Code)
Word 4:	Test Number	(BCD Code)

Figure 11. AMA Tape Format (Continued)

profile has been created the next word contains the malfunction set number for the new profile and the following word repeats the process starting at word 67. This information may require more than one physical record. After the multiple malfunction information has been completed the complete AMA information is repeated for a new step or sub-step of the test procedure.

The tape format for the malfunction set tape is shown in Figure 12. The first record is for users identification. The second record is the test procedure number for the malfunction information. The third record is either the AMA data for the static DI's in this procedure, in which case the format is identical to that of the data on the AMA tape, or the word "malfunctions". The next record and each succeeding record starts with a malfunction set number, and lists the number of components in the set and the names of the components. There are six RCA110 computer words reserved for each component name.

AMA EDITOR PROGRAM

Malfunction Set Tape Format

FIRST RECORD:

RCA 110A 16 Word identification Record

SECOND RECORD:

Word 1:	TEST	(BCD Code)
Word 2:	Test Number	(BCD Code)
Words 3-6:	Fill-in	

THIRD RECORD:

Either . . .

Information on DI's referenced in words 68-130 of the tape identification record on the AMA tape. This has the same format as the AMA data in record three starting with the DI profile

Or...

Malfunction set data

Words 1-3:	MALFUNCTIONS
Words 4-6:	Fill-in

Next Record: Malfunction Set

Each malfunction set will begin a new physical record, each of which has the following format:

Word 1:	Failure Set Number
Word 2:	Number of Components in Set
Word 3:	Component Name
Word 9:	Component Name
Word 15:	Component Name

Figure 12. Malfunction Set Tape Format

3/DISCRETE NETWORK SIMULATION MODEL

3.1 ENGINE CUTOFF SYSTEM

To demonstrate the capabilities of Discrete Network Simulation, a model of the S-1C Egnine Cutoff System (ECO) and the associated GSE was prepared. This model represents the equipment documented on 55 sheets of the advance electrical schematics. Table I lists the sheets used. The model includes 3043 variables as listed in Table II.

All variables in the model that do not change state during normal operation are made incative by placing an I in Column 73 of the variable Time Card. Such variables include diodes, fuses, resistors and closed switches. These variables remain as failure candidates for the Automatic Malfunction Analysis. The size of the operating model is further reduced by removing the time cards for most pins, reverse legs and some other variables which are not necessary for this models use with AMA. This leaves these variables in the original listing for documentation but effectively removes them from any additional processing. The variables used in equations but without their time cards removed are classified as unspecified in the DT&C Program.

After all these operations were completed, the model was simulated with 2425 active variables, 618 inactive variables and approximately 3000 unspecified variables.

3.2 GSE POWER DISTRIBUTION

The Electrical Power System Model was written to include power supply No. 1 and No. 2, the main GSE power supply, main Bus Distribution and all circuitry involved in power Bus generation. All of the system Bus Distribution was simulated, but certain busses were not applicable to TP60, TP01 or TP10. Consequently, these Busses were classified as "Inactive" for AMA but remain a portion of the model. Reverse legs and variables that do not change state during normal operation were classified inactive, the same as similar variables in the ECO Model. The model includes the electrical circuitry documented on the 90 sheets listed in Table III. Portions of the circuitry on Drawing 60B55701 (S-1C Stage Electrical Schematics) sheets 15, 16, 18 and 28 were included in the model to provide continuity of the GSE Power Distribution.

The model size is comparable to the ECO Model with a total of 3,240 variables. As shown in Table IV time cards were not included for connector pins and therefore they were classified as unspecified by the DT&C Program.

TABLE I-A

S1C NETWORKS IN DNS MODEL

DRAWING 60B55701

S1C STAGE ELECTRICAL SCHEMATICS

ENGINE CUTOFF NETWORKS

Sheet	
34	Inboard Engine Cutoff System
35	Outboard Engine Cutoff System
36	Outboard Engine Cutoff System
37	Cutoff Circuitry Engine No. 1
38	Cutoff Circuitry Engine No. 1
39	Cutoff Circuitry Engine No. 2
40	Cutoff Circuitry Engine No. 2
41	Cutoff Circuitry Engine No. 3
42	Cutoff Circuitry Engine No. 3
43	Cutoff Circuitry Engine No. 4
44	Cutoff Circuitry Engine No. 4
45	Cutoff Circuitry Engine No. 5
46	Cutoff Circuitry Engine No. 5
47	LOX Prevalves
48	Fuel Prevalves
49	Fuel Prevalve Position Indication
50	LOX Prevalve Position Indication
51	Stage Sequencing Switch Selector
52	Stage Sequencing Switch Selector

TABLE I-B

DRAWING SK65B74000

ADVANCED ELECTRICAL/MECHANICAL SCHEMATICS

Sheet	
399B	Pneumatic Supply Unit 401
445	Upper Stage Substitute Electrical Networks
449B	Switch Selector Control
555	GN_2 Pressurization
555A	GN_2 Pressurization
557A	GN_2 Pressurization
558	GN_2 Pressurization
558C	GN_2 Pressurization
568	Prevalves Control & Monitor, All
569	Prevalves Control & Monitor, All
569A	Prevalves Control & Monitor, All
570	Prevalves Control & Monitor, All
615	Rough Combustion/EDS
615A	Rough Combustion/EDS
615B	Rough Combustion/EDS
615C	Rough Combustion/EDS
615D	Rough Combustion/EDS
616	Engine Malfunction
617	Engine Malfunction
618	Engine Malfunction
618A	Thrust OK
619	Thrust OK
619B	Thrust OK
619C	Thrust OK
620	Thrust OK
620B	Thrust Not OK
620C	Thrust Not OK
621C	EDS Fuel & LOX Level & Pressure Cutoff
621D	EDS Fuel & LOX Level & Pressure Cutoff
621E	EDS Fuel & LOX Level & Pressure Cutoff
621F	LOX Level Cutoff Sensor Reset
622	Engine Cutoff Circuits
624	Engine Cutoff Circuits
643F	Engine Cutoff Circuits, Simulated Flight

TABLE II

DISTRIBUTION OF VARIABLES IN DNS MODEL OF S1C ENGINE CUTOFF SYSTEM

VARIABLES	ACTIVE	INACTIVE
DI (Discrete Inputs)	130	
Nodes	495	19
Coils	160	9
Busses	64	
Pins	245	16
Switches	53	55
Sensors	38	9
Channels	16	
Timers	11	
Inputs	19	8
DØ (Discrete Outputs)	59	1
Flight Comb. Monitor	15	
Solenoids	14	2
Valves	30	2
Regulators	2	2
Contacts	296	12
Legs	757	128
Diodes		185
Fuses		149
Resistors		20
Misc.	21	7
TOTAL	2425	624

TABLE III

DRAWING SK65B74000

ADVANCED ELECTRICAL/MECHANICAL SCHEMATICS

POWER DISTRIBUTION NETWORKS

Sheets	
140A	Bus 21D111
141 & 141A	Bus 21D117 & Bus 21D117A
161	Bus 21D116 & Bus 21D116A
163 & 163A	Measuring Rack Power Bus 1D23
165	Bus 21D118 & Bus 21D118A
188	Bus 21D114 & Bus 21D114A
269A thru 273E	Bus Distribution S1C GSE
277 thru 281	GSE DC Power Supply, Bus 21D110
287 thru 291	Power Supply No. 1, Bus 21D110
292A thru 296	Power Supply No. 2, Bus 1D210
297 thru 297C	Power to Vehicle, Bus 1D119
305 thru 310	Voltage Supervision
312	Bus 21D112 & Bus 21D112A
313	Power Change Over
448	Bus 21D115 & Bus 21D115A
451 & 451A	Bus 21D113 & Bus 21D113A
517	Bus 21D125 & Bus 21D125A
531	Bus 21D120 & Bus 21D121
534	Bus 21D122
535	Emergency Stop
549	Umbilical Separation Indication
559	GN_2 Vent Valve, Indication
572 thru 576	LOX Valves, Indication
577 thru 578	Main Fuel Valves, Indication
579 thru 580A	Gas Generator Valves, Indication
597	LOX Loading, Indication
607	Preparation Complete, Indication

TABLE IV

DISTRIBUTION OF VARIABLES IN DNS MODEL GSE POWER DISTRIBUTION SYSTEM

VARIABLES	ACTIVE	INACTIVE
DI (Discrete Inputs)	117	
Nodes	413	1
Coils	231	1
Busses	208	330
Pins	144	
Switches	68	
Sensors	16	
Inputs	170	3
Contacts	372	7
Legs	584	220
DO (Discrete Outputs)	31	
Valves	58	
Solenoids	4	
Motors	3	
Diodes		112
Fuses		141
Resistors		5
Meter		1
TOTAL	2419	821

Final AMA Data was obtained for the GSE Power System described with 2,419 active variables, 831 inactive variables and approximately 2,846 unspecified variables.

3.3 FUTURE APPLICATIONS

To provide the basis for additional application of the simulation program, the equations describing other networks were written but were not validated with the DNS Program. The networks by page number and a breakdown of the variables are listed in Table V.

TABLE V

SYSTEM NETWORKS FOR WHICH EQUATIONS WERE WRITTEN BUT NOT VALIDATED

Switch Selector with 427 Active Variables:

Drawing SK65B7400

Sheet	
449	Switch Selector Control
449A	Switch Selector Control
449B	Switch Selector Control
449C	Switch Selector Control
449D	Switch Selector Control
550	Switch Selector Control

Drawing 60B55701

51	Stage Sequencing Switch Selector
52	Stage Sequencing Switch Selector

Range Safety and Separation with 709 Active Variables:

Drawing SK65B7400

188	Manual-Auto Control Range Safety
189	Range Safety and Ordnance System
190	Range Safety and Ordnance System
190A	Range Safety and Ordnance System
191	Range Safety and Ordnance System
192	Range Safety and Ordnance System
192A	Range Safety and Ordnance System

Drawing 60B55701

53	Range Safety System Number 1
54	Range Safety System Number 2
55	Separation System

TABLE V Con't

Airborne Power Distribution with 249 Active Variables

Sheet	Drawing 60B55701
15	Batt. Power, Changeover Reg. & Monitoring
16	Seq. & Cont, Prpln & Timer Distributor
17	Measuring Distributors
18	Measuring Dist. & 5 Volt Measuring Supplies
19	5 Volt Distribution System
20	5 Volt Distribution System
21	Telemetry Power
22	Telemetry, Tape Recorder & Odop Power

Simulate Flight Systems

Sheet	Drawing SK65B74000
178	Terminal Countdown Sequencer
179	Terminal Countdown Sequencer
195	Range Safety and Ordnance System
282	75 KVA Motor Generator Set
282A	75 KVA Motor Generator Set
508	Ground Cooling
519	Hydraulic Supply System
538A	Pneumatic Supply System
543	Launch Simulator Bus Distributor
548	Plug Supervision
548A	Plug Supervision
549B	Simulated Umbilical Separation
550A	Thermal Conditioning Purge
551A	LOX Dome Gas Generator Purge
557	GN2 Pressurization
566	Hydraulic Checkout Valves
566A	Hydraulic Checkout Valves
570A	All Prevalves Monitor
587B	Turbo Pump Heaters
589	LOX Vent Valves Post Test
591	Hypergol Inst. Ind.
593	Fuel Loading
594	Fuel Loading

TABLE V Con't

Simulate Flight Systems:

Drawing SK65B74000

Sheet	
606	Arm Circuits
607A	Terminal Countdown Interface
614	Ignition Circuitry
620	Simulated Flight
621B	Simulated Flight
621F	Simulated Flight
626A	Simulated Flight
643A	Simulated Flight
643D	Simulated Flight
643E	Simulated Flight
643F	Simulated Flight

4/ SIMULATION OF THE MODEL

4.1 ENGINE CUTOFF NETWORKS

Drive functions were prepared for the simulation of the Engine Cutoff System based on the checkout test procedure, TP60. This procedure uses output signals (DO's) from the computer to substitute for stage generated signals and to control the GSE. The computer scans the available indicators (DI's) and compares them to a predicted value, any difference between the actual and predicted value causes a 'No Go' on the computer.

The test procedure is divided into blocks which are further divided into steps and sub-steps. TP60 is divided into 19 blocks as shown in Table VI. Figure 13 is a typical page from this test procedure.

Before starting a test, the hardware must be in the same state it would be during operation; the right busses energized, switches in correct position, etc. In the model these same requirements apply and are accomplished by inputting these variables to the desired state at the correct time. When the state of all variables are correct, the DO's (Discrete Outputs) are input in the same order and in the same relative time as in the test procedure. After the DO has been input and at the same relative time as the scan in the test procedure, a list is made of the state of all variables in the model. As the test procedure monitors only Discrete Inputs, a variable print option in the simulation program is used, causing the list printed to contain only DI's and DO's. This allows an easy check against the predicted state of the DI's and DO's as given in the Test Procedure; however, the Binary output tape contains the complete state list of all model variables. Figure 14 is a copy of a page of the simulation printout produced from the model simulation of that part of TP60 shown in Figure 13. This simulation was in the variable print mode and only DI's and DO's printed.

4.2 GSE POWER DISTRIBUTION SYSTEM

Test Procedure T-1 is used to turn on the power to the GSE, and T-10 controls the GSE to supply power to the stage. Both of these procedures were used as the source of the drive functions for the GSE Power Distribution System simulations using the same method as was used with the Engine Cutoff Networks.

TABLE VI

TP60 is divided into 19 blocks which are further made up of steps and sub-steps. Each block covers some specific part of the test as follows:

Block Number	Sub System Tested
BØTB000	Initialization
001	Instrument Unit Cutoff Test
003	Engine No. 2 Thrust OK Sensor Test
004	Engine No. 3 Thrust OK Sensor Test
005	Engine No. 4 Thrust OK Sensor Test
006	Egnine No. 5 Thrust OK Sensor Test
007	Egnine No. 1 Flight Combustion Sensor Test
008	Engine No. 2 Flight Combustion Sensor Test
009	Engine No. 3 Flight Combustion Sensor Test
010	Engine No. 4 Flight Combustion Sensor Test
011	Engine No. 5 Flight Combustion Sensor Test
012	GSE Cutoff and Automatic Abort Test
013	Outboard LOX Level Cutoff Sensor Test
014	Outboard Fuel Cutoff Sensor Test
015	Inboard Engine LOX Level Cutoff Sensor Test
016	Inboard Engine Fuel Level Cutoff Sensor Test
017	Inboard Engine Cutoff Test
BSDN008	Shutdown

REVISIONS			
SYM	DESCRIPTION	DATE	APPROVAL

STEP NUMBER		OPERATOR	OPERANDS							REMARKS
			FIXED						VARIABLE	
STEP	SS		CON	VALUE	LOWER LIMIT	UPPER LIMIT	UNIT	TIME		
		BOTB							001	INSTRUMENT UNIT CUTOFF TEST
1	05	MESG		G20	2	1	BR		28,INSTRUMENT UNIT CUTOFF TE	
			C						ST	
	06	DELY						10000		
	07	MESG		ERASE	2					
6	05	DISQ	1						D0786	DEE 294
										EDS
										INSTRUMENT UNIT ENGINE AND
										PREVALVE CUTOFF COMMAND
	10	DISI	1						DI1217,DI1218,DI1219,DI1220,	DEE 243,244,245,246,247,57,
			C						DI1221,DI420,DI15,DI119,DI8,	53,54,55,56
			C						DI11	EDS
										ENGINE CUTOFF INDICATIONS
	35	DISI	1						DI422,DI424,DI426,DI428,DI43	DEE 026,024,026,034,030,032,
			C						0,DI384,DI386,DI388,DI390,DI	040,036,038,046,042,044,052,
			C						392,DI394,DI396,DI398,DI400,	048,050
			C						DI402	
	40	DISI	0						DI423,DI425,DI427,DI429,DI43	DEE 027,023,025,033,029,035,
			C						1,DI389,DI391,DI393,DI395,DI	037,045,041,043,051,047,049,
			C						397,DI399,DI401,DI403,DI387,	039,180,031
			C						DI385,DI404	EDS

E04-66B10060

SHEET 9

Figure 13. Sheet from Engine Cutoff Test Procedure

TP60 ECØ TEST MØDEL FØR AMA PAGE 229

TYPE	DESCRIPTIØN	VALUE	NØ. ØF EVENTS	DAYS	HØURS	MINUTES	SECØNDS
*TRANSLATIØN MØDE							
TYPE	DESCRIPTIØN	VALUE	NØ. ØF EVENTS	DAYS	HØURS	MINUTES	SECØNDS
*BEGIN,4000.	BLØCK 1, IU CUTØFF TEST						
INPUT	BYE (DO786)	1	1				5.000
ENTER	ACV (DI 1217)	1	21				5.015
ENTER	ACW (DI 1218)	1	20				5.015
ENTER	ACX (DI 1219)	1	19				5.015
ENTER	ACY (DI 1220)	1	18				5.015
ENTER	ACZ (DI 1221)	1	17				5.015
ENTER	ADA (DI 1223)	1	16				5.015
ENTER	AAC (DI 15)	1	15				5.015
ENTER	AAW (DI 119)	1	14				5.015
ENTER	AAA (DI 8)	1	13				5.015
ENTER	AAB (DI 11)	1	12				5.015
ENTER	ACC (DI 420)	1	11				5.015
ENTER	ACG (DI 425)	0	31				5.020
ENTER	ACI (DI 427)	0	30				5.020
ENTER	ABM (DI 389)	0	29				5.020
ENTER	ABØ (DI 391)	0	28				5.020
ENTER	ABS (DI 395)	0	27				5.020
ENTER	ABU (DI 397)	0	26				5.020
ENTER	ACE (DI 423)	0	25				5.020
ENTER	ACK (DI 429)	0	24				5.020
ENTER	ACM (DI 431)	0	23				5.020
ENTER	ABI (DI 385)	0	22				5.020
ENTER	ABK (DI 387)	0	21				5.020
ENTER	ABQ (DI 393)	0	20				5.020
ENTER	ABW (DI 399)	0	19				5.020
ENTER	ABY (DI 401)	0	18				5.020
ENTER	ACA (DI 403)	0	17				5.020
ENTER	ACB (DI 404)	0	16				5.025
ENTER	ACF (DI 424)	1	15				5.520
ENTER	ACH (DI 426)	1	14				5.520
ENTER	ABL (DI 388)	1	13				5.520
ENTER	ABN (DI 390)	1	12				5.520
ENTER	ABR (DI 394)	1	11				5.520
ENTER	ABT (DI 396)	1	10				5.520
ENTER	ACD (DI 422)	1	9				5.520
ENTER	ACJ (DI 428)	1	8				5.520
ENTER	ACL (DI 430)	1	7				5.520
ENTER	ABH (DI 384)	1	6				5.520
ENTER	ABJ (DI 386)	1	5				5.520
ENTER	ABP (DI 392)	1	4				5.520
ENTER	ABV (DI 398)	1	3				5.520
ENTER	ABX (DI 400)	1	2				5.520
ENTER	ABZ (DI 402)	1	1				5.520

THERE WERE 477 EVENTS IN CASE 14.

Figure 14. Simulation Printout for Test Procedure

5/AUTOMATIC MALFUNCTION ANALYSIS APPLICATION

The AMA Data developed by General Dynamics Convair utilizes the 7090/94 Computer and stores the information on magnetic tapes that are compatible with the RCA 110A Computer. The necessary program to utilize these tapes on the RCA 110A during checkout was prepared by MSFC-R-QUAL and the Boeing Company. Figure 15 indicates the data flow for the AMA process. The model (Boolean equations) is the input to the Down Translation and Culling Program which condenses and formats the engineering data for processing in the computer. The Preprocessor Editor Program analyzes the equations to establish the interrelationships of the model used in both the Simulation and AMA programs. The Simulation Program, with inputs representing the test procedure, produces at each analysis point a 'state list' (the status of all the active variables in the model). The AMA Program analyzes the system starting at each indicator to determine the possible failure candidates. The input from the Down Translation and Culling Program contains the classification of the variables. The tape from the Preprocessor Editor Program defines the interrelationship of the variables in the model and the output from the Simulation Program gives the state list which is the basis for analysis. The AMA Editor Program formats the generated data for efficient information storage on tape. The RCA 110A Checkout Computer searches these tapes and presents the component failure candidates on the display console.

There are two tapes mounted on the RCA 110A for AMA data, the search information by block, step and sub-step is on one, and the Malfunction Sets are on the other. AMA data may be retrieved during checkout if a 'No-Go' occurs at a SCAN operator. If the test procedure is not at a SCAN operator an error message, "This Option is available only at SCAN operator", will be displayed.

When there is more than one (1) DI that is 'No-Go' the test conductor may request multiple or single DI failure information. If the multiple fault isolation option is selected, the programs will start comparing AMA generated DI profiles with actual system DI profiles, and continue until an AMA pattern of DI's are located that fall within the limits of the actual system profile.

The AMA profile with the maximum number of 'No-Go' DI's will be selected first. The test conductor may initiate additional searches for other DI Patterns. When there are no multiple DI profiles that can match the actual profile, the message, "No more multiple malfunctions available at this point" is displayed.

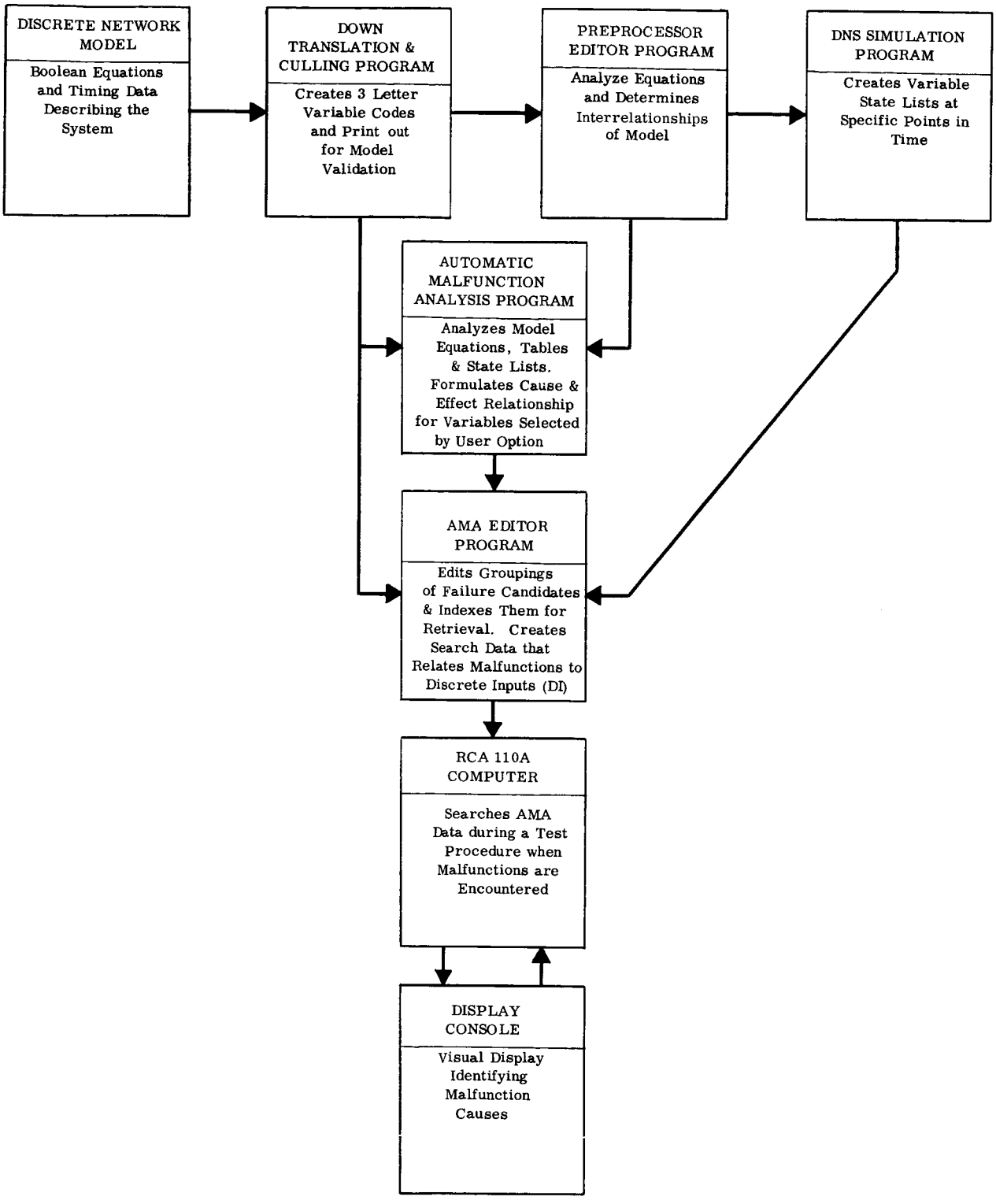


Figure 15. Automatic Malfunction Analysis Technique

AMA data furnished by General Dynamics Convair was generated to contain only the data that is actively connected to the S-1C Engine Cutoff System. Therefore, it will only provide AMA data for the specific discretes related to the Engine Cutoff System Test Procedure.

During either option (Multiple or Single) the 'No-Go' DI's will be compared to a reference profile on the AMA tape to see if AMA data is available for the DI's. If no data is available, a message, "No AMA Data available for DI" is displayed.

Once a single DI is selected, the actual state of the DI is compared to the model simulation state as recorded on the AMA tape for that sub-step; if the two states agree, a message, "Either the Test Procedure or the Model is in error for DI", is displayed.

After the valid 'No-Go' is established, the "Set No." for the DI becomes the search key for the Malfunction Set Tape. The set tape is searched and the components (model variables) that could cause the selected DI to be 'No-Go' are indicated on the Display console. When the "Multiple" option is used the DI's that are 'No-Go' in the AMA profile are also displayed. After the displayed information has been analyzed the Test Conductor may request information on additional DI's or exit the fault isolation routine and return to the Test Procedure.

Figure 16 shows a test console CRT monitor display for a single DI search. The information listed on the top portion of the CRT display identifies the test procedure, test block, test step, and test sub-step. Immediately following is the AMA calling routine, "XF13", and the title of the AMA Program. The next line displayed lists the discrete that is under analysis, and the "Normal" or "AMA Predicted" state of the discrete at the time of the AMA analysis. In this example it shows DI 86 and lists its normal state as "On". Directly following are all the variables that individually could effect DI 86, and could cause it to be "Off" instead of "On" at this point in the test procedure. This group of listed variables constitutes the "Malfunction Set", that was formulated during the five DNS/AMA Programs. Figure 17 is a simplified schematic of the circuit that terminates with DI 86. The circuit shown is a serial path and any interruption of continuity would cause an "Off" indication of DI 86. All of the variables listed are portions of this serial path, and individually could cause a failure. In this case, it is not possible to isolate the malfunction in greater detail without adding additional monitoring points.

The BUS115A91D119 has an actual state of "On", and has been eliminated as a failure candidate because it's failure would not be unique to DI 86 in this case.

The simplified circuitry for DI 1355 shown on Figure 18 includes the variables displayed in the Malfunction Set photograph on Figure 19. All of the variables shown with the exception of Cont 115A9K31C1011 can be eliminated as failure candidates at this time because of their AMA predicted state was "On", ('1') and the DI 1355 actually was erroneously "On".

PRØCEDURE T60A STATUS
BLCK 000 STEP 076 SUB-STEP
FAULT ISØLATIØN RØUTINE XFI3.
AUTØMATIC MALFUNCTIØN ANALYSIS ØF DISCRETES

MALF SET DATA FØR DI 0086 WITH AMA STATE ØF ØN

LEG2N5A6J17SZ
FUSE4A3-53
CØNTI15A9K11C46
CØNTI15A9K21C79
CØNTI15A9K31C79
LEG3N6A4J18G

DØ YØU WISH TØ EXIT THIS FAULT ISØLATIØN RØUTINE

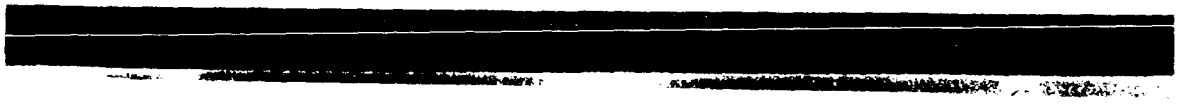
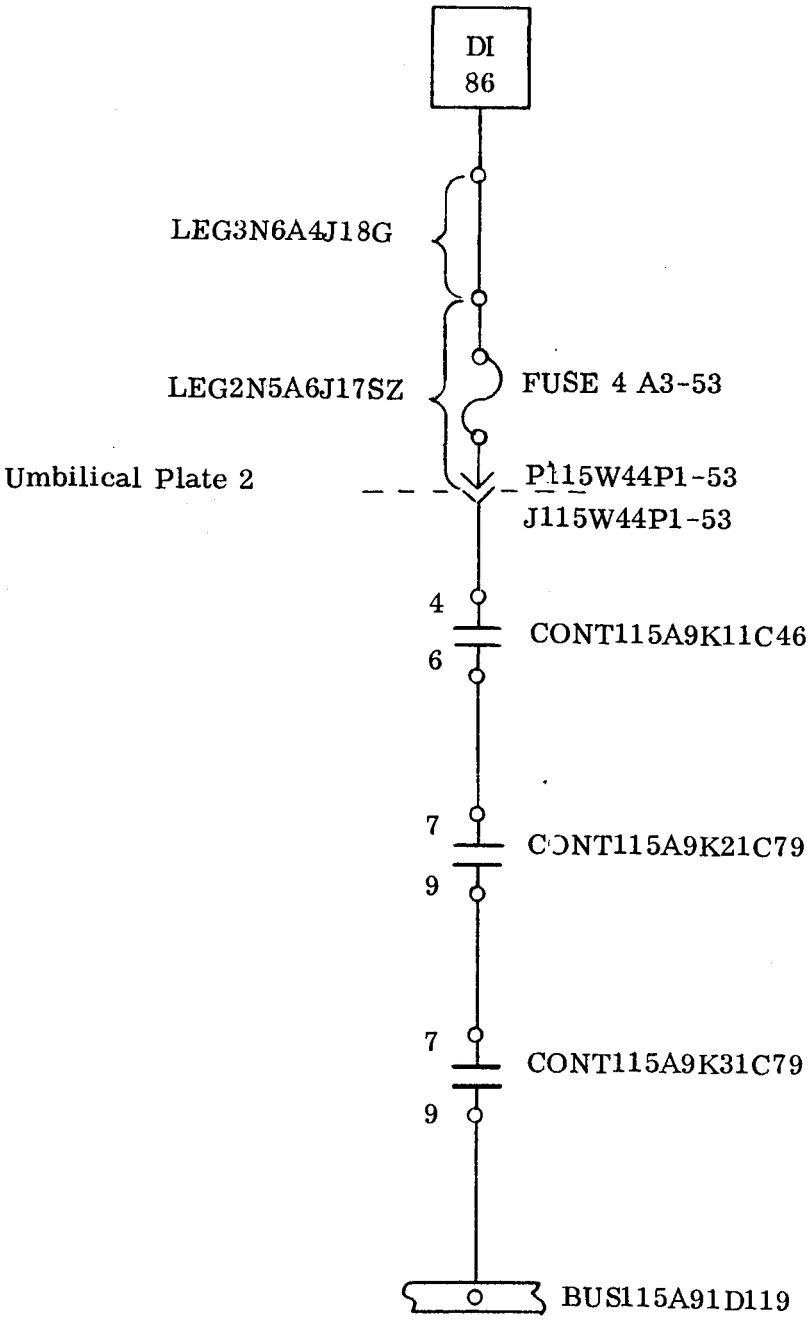
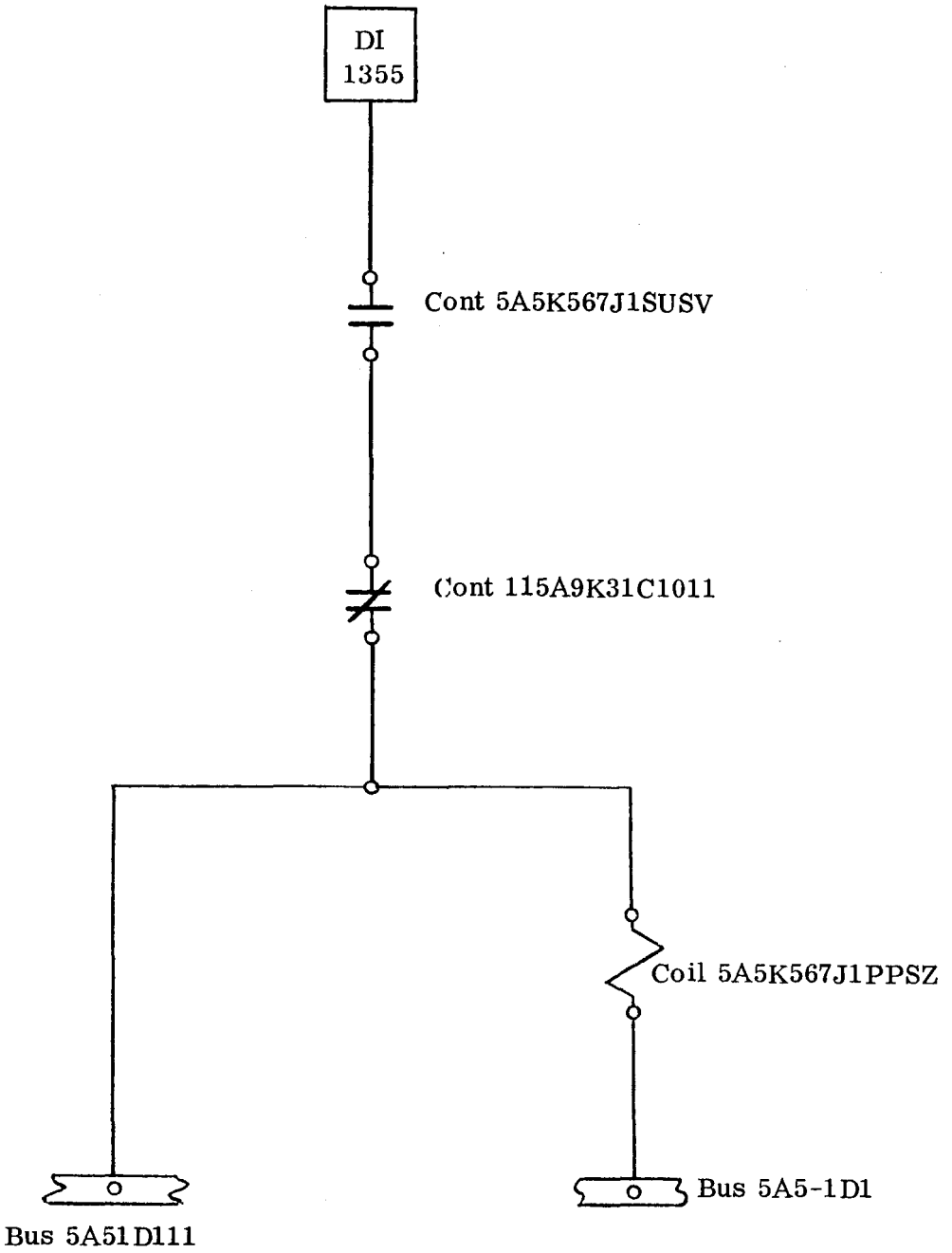


Figure 16. AMA Display for DI 86



For Detailed Circuitry, See Saturn S1C Drawing 60B55701 Sheet 37 and Saturn S1C GSE Drawing SK65B74000 Sheet 618A.

Figure 17. Simplified Flow Schematic, DI 86



For Detailed Circuitry see Saturn S1C Drawing SK65B74000 Sheet 620B and Drawing 60B55701 Sheet 43.

Figure 18. Simplified Flow Schematic, DI 1355

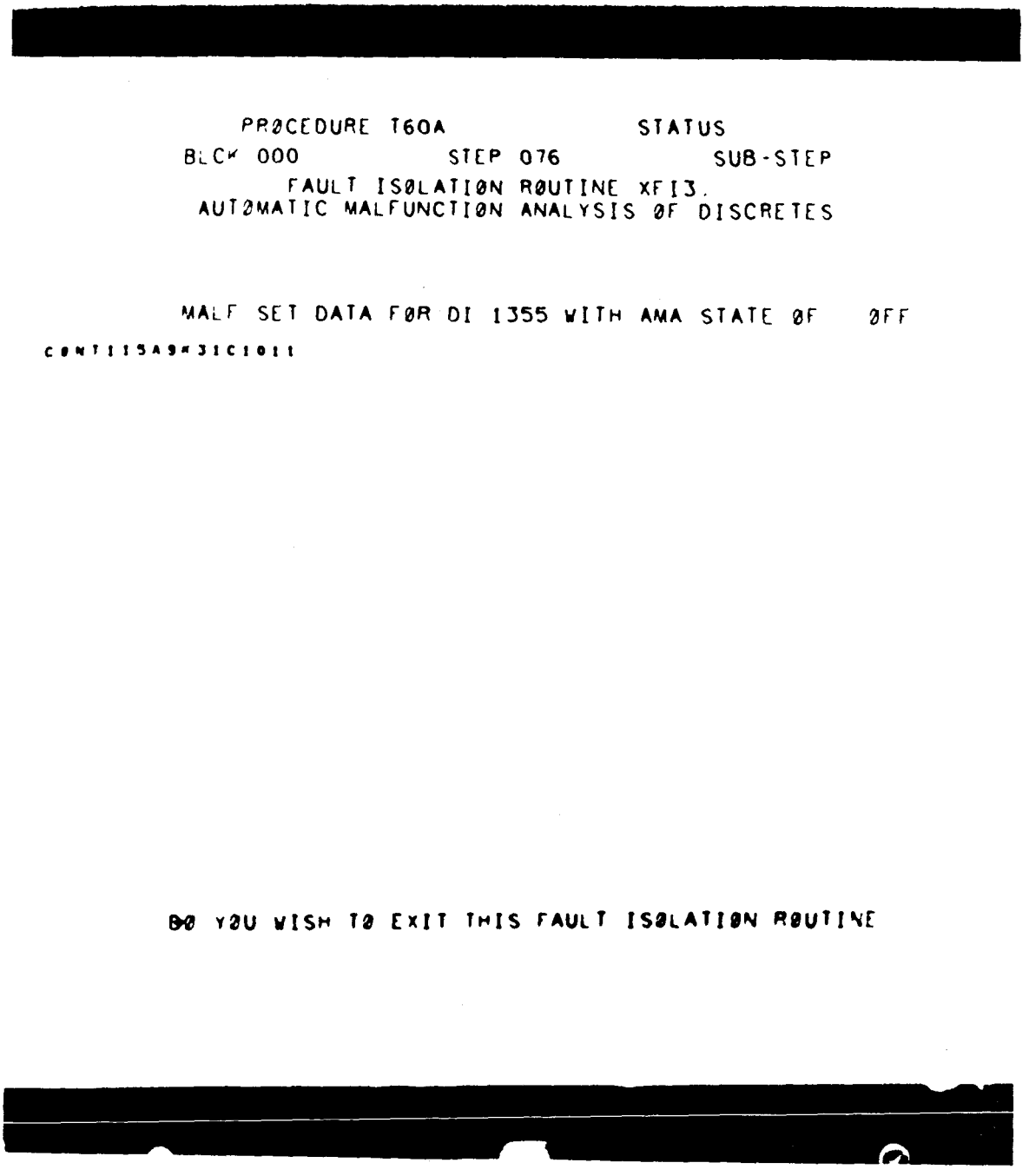


Figure 19. AMA Display for DI 1355

Contact 115A9K31C1011 is normally closed and has an AMA predicted state of "Open". Therefore, it is the only singular variable in this circuit whose failure could supply power to DI 1355 at this time. The Coil 115A9K31 (not shown) was not considered as failure candidate since its failure would affect other sets of its contacts, which in turn would affect other DI's. If the analysis had been for multiple DI failures the analysis would have continued until the variable or variables had been located, that could cause a particular pattern of DI's to be 'Off' or 'On'.

Figure 20 shows the "Malfunction Set" responsible for the failure of DI 8, at the step of the procedure shown. The circuitry involved is illustrated in Figure 21. The AMA predicted state was 'Off', therefore, the 'No-Go' was 'On', indicating an unpredicted source had supplied power to DI 8. The power Bus and the return Bus were again eliminated by AMA as failure possibilities because their predicted state was "1", and failure of either would not be unique to DI 8. The DIODE and the fuse could only fail in a manner which would cause the circuit to be "Open", so it is apparent that either of the two remaining normally open contacts, if closed, would have supplied power to DI 8.

The explanations of the preceding examples were based on visual logic rather than AMA Program procedures. This was done to simplify the overall Automatic Malfunction Analysis concept for those not familiar with programming details.

Figure 22 is a photograph of a typical multiple DI malfunction CRT display. BUS115A91D119 is the malfunction candidate and is the only variable included in malfunction set no. 464. The DI's in the AMA profile that caused this malfunction set to be selected are shown at the bottom of the display.

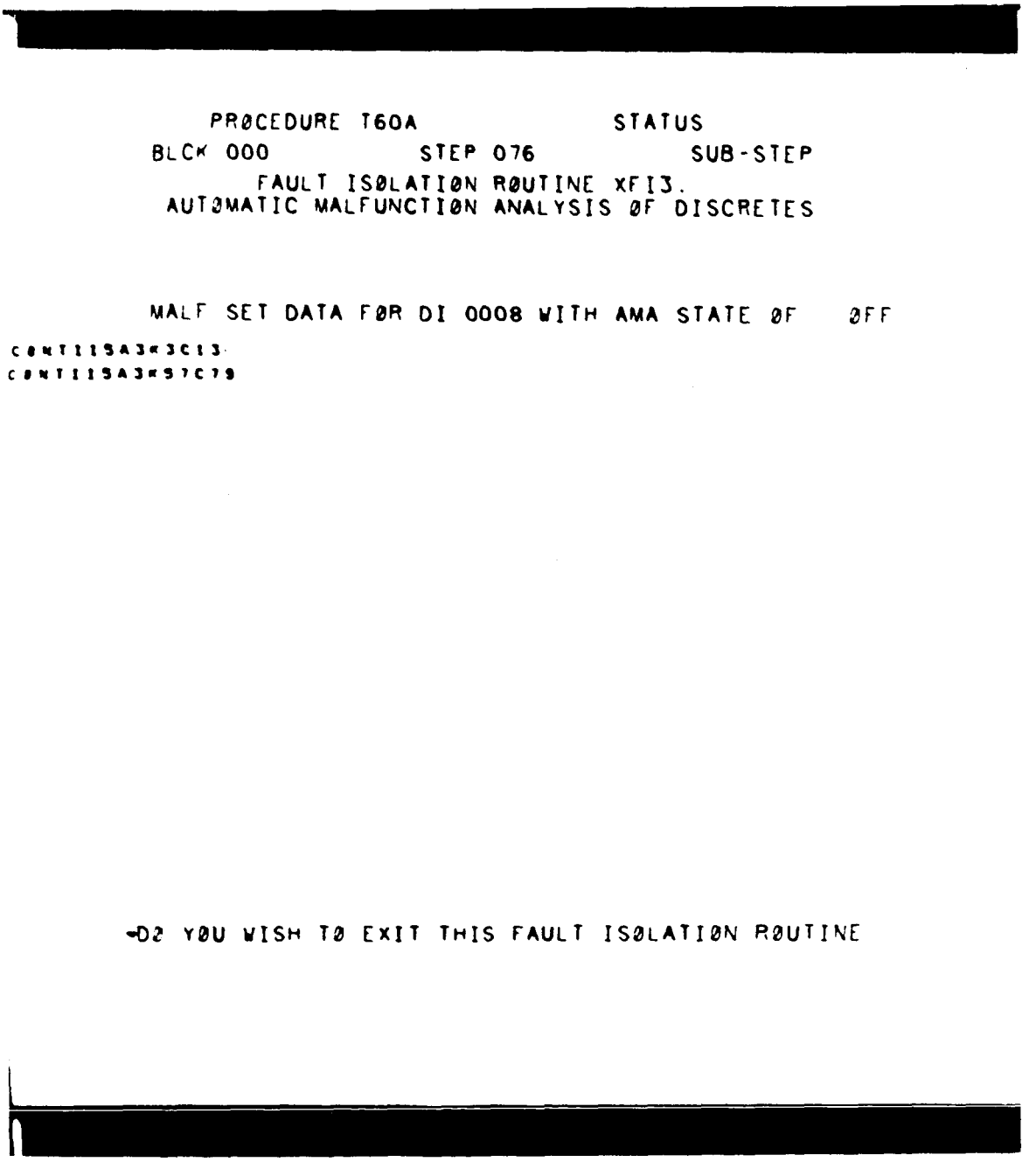
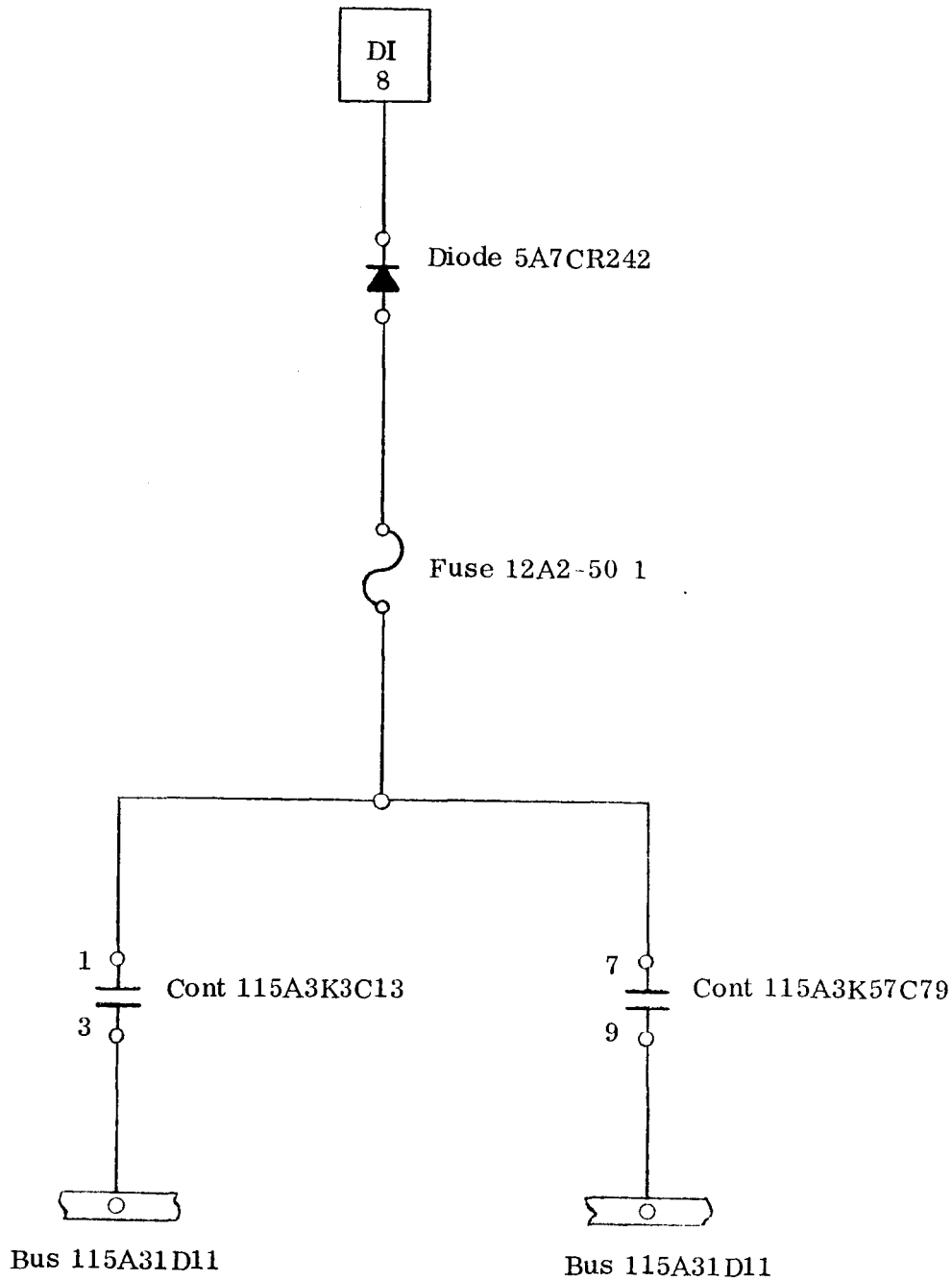


Figure 20. AMA Display for DI 8



For Detailed Circuitry see Saturn S1C Drawing SK65B74000 Sheet 622 and Drawing 60B55701 Sheet 37.

Figure 21. Simplified Flow Schematic, DI 8

PRØCEDURE T60A STATUS

BLCK 000 STEP 076 SUB-STEP 40

FAULT ISØLATIØN RØUTINE XFI3.

AUTØMATIC MALFUNCTIØN ANALYSIS ØF DISCRETES

DØ YØU WISH TØ CHECK FØR FURTHER MULTIPLE MALFUNCTIØNS

MULTIPLE MALFUNCTIØN SET DATA AND DISCRETES

SET NØ. 0464

BUS115A91D119

0086
0087
0088
0089
0090

Figure 22. AMA Display For Multiple DI Search

6/CONCLUSIONS

Discrete Network Simulation is an analytical tool that can be applied to several problems common to large systems once the logic model has been built.

1. Simulation can be used to verify that the test procedure stimulated the hardware correctly.

2. Simulation can evaluate the effects of engineering changes before they are incorporated.

3. The data generated by the DNS Programs can be used to estimate the adequacy of the monitoring system to detect critical failures.

4. Simulation can determine the operating time of the complete system.

Automatic Malfunction Analysis is a specific application of DNS as applied to the support of automatic checkout operations.

General Dynamics has been able to generate AMA data and with MSFC assistance been able to demonstrate the accuracy of the data. NASA as the potential user must evaluate the value of AMA data. However, before this can be done, the criteria for the evaluation must be established. The value of AMA data is a function of the familiarity that the checkout personnel has with the hardware. Thus, AMA has its minimum value when used in support of the first vehicle when highly trained and motivated crews are available. This value increases as normal attrition and program expansion results in less familiar crews being used.

AMA data can provide an immediate answer, during test to the possible failure candidates in the event of a failure. This allows a decision as to what action to take. During the checkout and launch operation, the time required for such decisions becomes critical. Therefore, it is the launch operation that would benefit the most from the application of AMA data to the complete Saturn V vehicle. If the use of AMA technique could prevent the abortion of one scheduled launch, the complete cost of developing AMA would be justified.

7/REFERENCES

1. "Simulation of Selected Discrete Networks," NASA Contract No. NAS8-20016, Phase I Final Report, GDC DDF65-005, Volume One, "The Logic Model," General Dynamics/Convair, Huntsville, Alabama, September, 1965.

2. "Simulation of Selected Discrete Networks," NASA Contract No. NAS8-20016, Phase I Final Report, GDC DDF65-005, Volume Two, "Users Reference Guide Discrete Network Simulation Program," General Dynamics/Convair, Huntsville, Alabama, September, 1965.

3. "Simulation of Selected Discrete Networks," NASA Contract No. NAS8-20016, Phase I Final Report, GDC DDF65-005, Volume Three, "Saturn I-C Engine Cutoff System Model," General Dynamics/Convair, Huntsville, Alabama, September, 1965.

4. "S-1C Stage Electrical Schematics," Drawing 60B55701, National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama.

5. "S-1C Ground Support Equipment Electrical/Mechanical Schematics," Drawing SK65B74001, National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama.

6. "Engine Cutoff Systems Test Procedure," T60 66B11060 E07, National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama, May, 1966.

7. "GSE Electrical System Start-up," T01 66B11001 A 03, National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama, June, 1966.

8. "Stage Electrical System Start-up," T10 66B11010 A, National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama, April, 1966.

APPENDIX A

"Simulation of Selected Discrete Networks " Phase I Final Report, Volume One, is included in this report as Appendix A.

GENERAL DYNAMICS | CONVAIR

PHASE I FINAL REPORT

SIMULATION OF SELECTED DISCRETE NETWORKS

VOLUME ONE

THE LOGIC MODEL
CONTRACT NAS8-20016

Report No. GD/C DDF 65-005

Prepared by

GENERAL DYNAMICS/CONVAIR
A Division of General Dynamics Corporation
Huntsville, Alabama

for

George C. Marshall Space Flight Center
Huntsville, Alabama

Revised September 1966

CONTENTS

ILLUSTRATIONS

TABLES

INTRODUCTION

The Discrete Network Simulation (DNS) system is based on simulation and analysis techniques developed for the Atlas Weapon System under government and corporate sponsorship. The total technique as applied to the Atlas Weapon System was called FASTI, Fast Access to System Technical Information. This study uses the Discrete Network Simulation portion with a modified version of the documentation and retrieval process. Digital computer programs are used to simulate discrete networks in less than real time. These programs were developed by GD/A and then incorporated into the FASTI system.

The prime purpose of Discrete Network Simulation methodology is to provide a set of analytical tools capable of conducting thorough, accurate and rapid analysis of complex systems. The methodology consists basically of:

A system network model.

A set of computer programs which will operate and activate the model.

These programs provide a realistic analysis and prediction of system performance before or after the hardware system is constructed. It is another form of testing; the results are as valid as those obtained by the more common hardware test procedures.

The Discrete Network Simulator (DNS) chronologically simulates events occurring due to the interactions among elements in a system network. Each "event," a Boolean change of state, is the result of a logical cause and effect relationship among elements in the system. The system modeled for the simulation may be a switching circuit, man/machine interaction, or any network where the component or subcomponent interrelations may be defined logically.

Convair is conducting a study under NASA Contract NAS8-20016, which applies the Discrete Network Simulation techniques to the Saturn SIC Engine Cutoff System networks. This report summarizes the results of Phase I and consists of three (3) volumes.

Volume One describes the methodology for constructing a DNS model.

Volume Two describes the DNS computer programs to the "Programmer." It is the "Users Reference Manual" for DNS.

Volume Three summarizes the study of the SlC Engine Cutoff System. The DNS Model and examples of the system simulation are described.

1/THE SYSTEM MODEL

The model is a series of Boolean algebra equations that logically describe the component interrelationships of a system network or partial system network. The Discrete Network Simulation programs (DNS) will chronologically simulate events occurring as a result of dynamic interactions among elements in a model. The use of the term, "Discrete Network," implies a system of variables defined in interdependent relationship. Each variable is discrete as its action is a binary event in the network: on or off, acting or not acting, available or not available, true or false, 1 or 0 etc. The variables represent events or activities. The network is described by: (1) a set of Boolean equations which completely define all interrelationships, and (2) a characteristic activity time associated with each variable (the time required for the effective change of state of a particular variable), which forms a complete mathematical model of a physical system.

1.1 DATA FOR MODEL BUILDING.

In order to develop a model, the following information must be available:

1.1.1 A complete set of circuit diagrams for the system to be modeled.

1.1.2 Information that shows mechanical to electrical ties and vice versa, such as limit and position switches for valves. If written description is not available, all the related mechanical drawings will be needed.

1.1.3 Design specifications for relays, automatic switches etc., where operating times are given.

1.1.4 An analysis prior to starting the model should be made to determine the boundary of the model, the number of variables that the model contains, and the inputs that will be necessary to make the model function.

1.1.5 The ground rules used to identify variables in the model can best be described by ref. to Figure 1-1, which is a portion of one of the Electrical Support Equipment drawings used to produce the model. The identifying names of the variables are divided, in general, into three parts as follows: (1) The engineering name by which

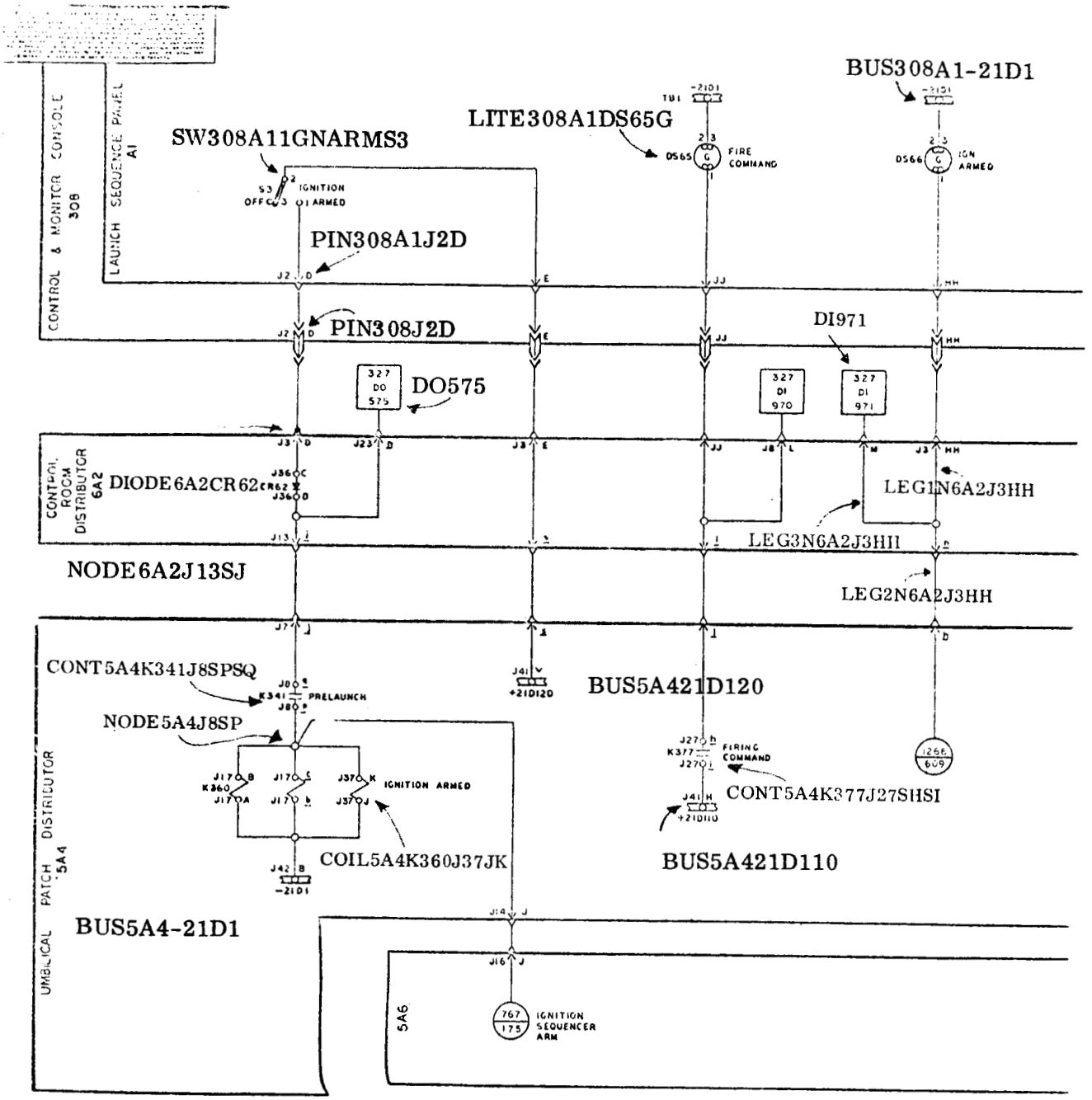

BUS308A1-21D1
LITE308A1DS65G
SW308A11GNARMS3
PIN308A1J2D
PIN308J2D
DI971
DO575
DIODE6A2CR62
LEG1N6A2J3HH
LEG3N6A2J3HH
LEG2N6A2J3HH
NODE6A2J13SJ
CONT5A4K341J8SPSQ
BUS5A421D120
NODE5A4J8SP
CONT5A4K377J27SHSI
COIL5A4K360J37JK
BUS5A421D110
BUS5A4-21D1
CONTROL & MONITOR CONSOLE 308
LAUNCH SEQUENCE PANEL A1
CONTROL ROOM DISTRIBUTOR 6A2
UMBILICAL PATCH DISTRIBUTOR 5A4
5A6
IGNITION ARMED
PRELAUNCH
FIRING COMMAND
IGNITION SEQUENCER ARM

FIGURE 1-1

the variable would normally be called such as, COIL, DIODE, LITE, NODE, LEG, etc. Names such as CONTACT, SWITCH, etc., are abbreviated to keep the variable length below 24 characters. (2) The second part of the variable will normally be the designation of the hardware unit in which the variable is located with the designation as shown on the drawing, such as 5A7K402, 6A4CR3, 400A12DS89 etc., (3) The final part of the variable will pinpoint the exact location of the variable to a pin or terminal in a particular chassis.

The above rules apply to the majority of variables. In all cases they will be followed as closely as possible, to make the variable recognizable by anyone using the simulation. A variable must always be designated exactly the same in every equation or time-card in a model.

The following table shows how variables in Figure 1-1 were named:

ENGINEERING NAME	HARDWARE UNIT AND DRAW. DESIG.	UNIQUE LOCATION TO PIN OR TERM.
COIL	5A4K360	J17AB
COIL	5A4K360	J17SBSC
COIL	5A4K360	J37JK
PIN	308A1	J2D
DIODE	6A2CR62	
LITE	308A1DS65G	
LITE	308A1DS66G	
NODE	6A2	J13SJ
NODE	6A2	J3JJ
CONT	5A4K377	J27SHSI
LEG1N	6A2	J3JJ
LEG2N	6A2	J3JJ
LEG3N	6A2	J3JJ
BUS5A4-21D1	5A4	
BUS5A421D120	5A4	
BUS5A421D110	5A4	
BUS308A1-21D1	308A1	

TABLE 1-1

*Where lower case letters are used to designate a pin, an "S" is placed before the upper case of the same letter to better identify them in the model as the keypunch does not have lower case letters.

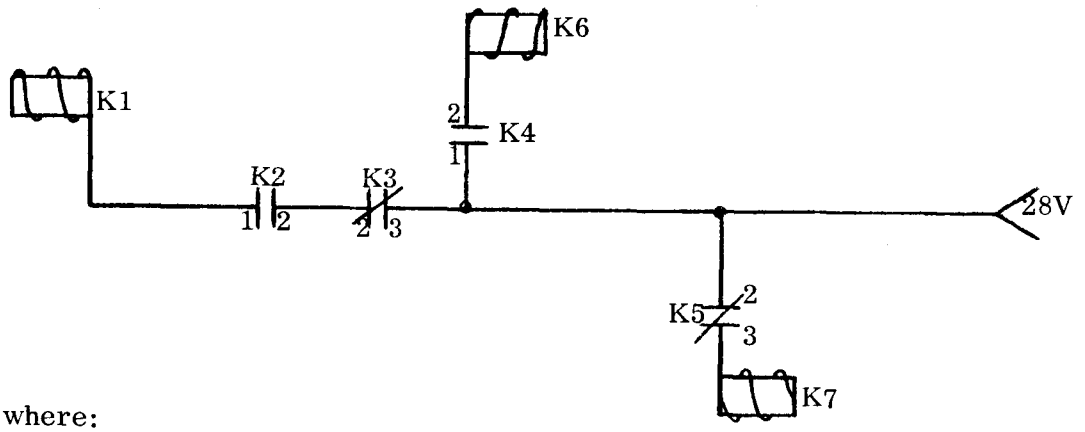
As can be seen from Table 1-1, all variables do not use every column of the allotted field. Discretes were not tied to any chassis. The DNS Program

does not allow a space between characters in a variable. There is no separation of the variables into three parts, as shown in the table.

2/BOOLEAN EQUATION AND TIME PARAMETERS

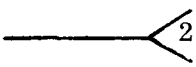
2.1 DEVELOPMENT OF BASIC BOOLEAN EQUATIONS USED IN A BASIC SIMULATION

Normally, all drawings show systems in a de-energized state. This is most evident with electrical drawings, for example:

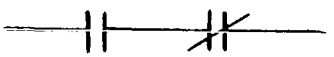


where:

-RELAY COIL



-VOLTAGE SOURCE



-RELAY CONTACTS

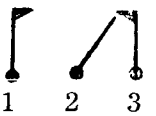


FIGURE 1-2

As shown, "1-2" contacts are open contacts when its relay coil is de-energized. Similarly, "2-3" contacts are closed contacts when its relay coil is de-energized. Therefore, to energize the above relay coil "Kl," "K2" and "28V" must be

energized, but not "K3." Stated again more discretely: Kl is energized when K2 is energized, K3 is not energized and 28V is energized.

2.2 SYMBOLS USED IN BOOLEAN EQUATIONS

The following basic symbols will be assigned for Boolean equations:

SYMBOL	MEANING
=	Equal
*	and
+	or
/	not

Restating the previous relationship: Kl is energized = K2 is energized * /K3 is energized * 28V is energized. The wording, "is energized," can be dropped, since it is common to all the variables in the equation, resulting in:

$$Kl = K2 * /K3 * 28V \quad (1)$$

which expresses all necessary actions required to energize "Kl." This is the basic technique common to all Boolean equations for the SIC Engine Cutoff System. The equations are written describing the variable in its de-energized state. This is the state that is shown on most drawings.

2.3 BINARY REPRESENTATION OF VARIABLES

Because every variable in the system can only be described as "energized" or "not energized" (VARIABLE or /VARIABLE, respectively) and since all variables are described in Boolean Algebra, there are only two possible states for a variable. In Boolean Algebra the two states are represented by a "l" and a "0." "One" represents a signal, or energized, and "0" represents a lack of a signal, or not energized.

2.4 OPERATION OF THE SIMULATION

In the system simulation, a variable may change from a binary "l" to a binary "0" or from a binary "0" to a binary "l." The cause of this change could be from either the actions of associated variables or from actions of commands in the INPUT. The INPUT is a listing of commands which at the time specified by the particular INPUT will enter the simulation causing other variables to react. Should there be any contradiction caused by this entry, the input command will dominate.

Given the following circuit:

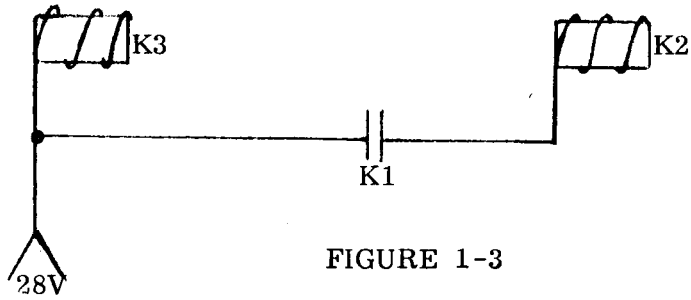


FIGURE 1-3

and the following INPUT commands:

28V = 1 at Time 12

28V = 0 at Time 259

the Boolean equations for "K2" and "K3" are:

$$K2 = K1 * 28V \quad (2)$$

$$K3 = 28V \quad (3)$$

In the system simulation, when the input command, "28V = 1 at Time 12," is applied to Eq. 2 and 3, action will be taken on Eq. 3, causing this variable to change from a binary "0" to a binary "1." There is no change to Eq. 2 since the variable K1 has not been energized. Should K1 later become a binary "1" before Time 259, Eq. 2 will then also become a binary "1."

At a later time prior to Time 259, should K1 become a binary "0," K2 will return to a binary "0." At time 259 when 28V becomes a binary "0," then K3 will become a binary "0."

2.5 TIME IN THE SIMULATION

In a system simulation there must be assigned to each variable a pickup and a dropout time, that is the time necessary for the variable to change its binary state. The pickup time is the time necessary for the variable to change its state from a binary 0 to 1, while the dropout time is the reverse. The pickup and dropout times are broken down further into minimum, average, and maximum, and when a simulation is made, any combination of these pickup and dropout times can be used. Therefore, if a relay, K1, has a nomimal pickup time of 20 ms with a tolerance of

± 10% and a nominal dropout time of 10 ms but also with a tolerance of ± 10%, the six times associated with this variable would be:

Minimum Pickup Time -- 18 ms

Average " " -- 20 ms

Maximum " " -- 22 ms

Minimum Dropout Time -- 9 ms

Average " " --10 ms

Maximum " " --11 ms

Having chosen the average time category and having written an equation describing the previous relay Kl (Figure 1-3), the pickup time is 20 ms. The Boolean equation stated for Kl was:

$$Kl = K2 * /K3 * 28V \quad (1)$$

When K2 and 28V are energized and K3 is not energized, Kl tries to become energized. This simulates the completion of the "copperpath" to the coil of relay Kl. However, due to the mechanical delay of the relay, its contacts do not change their position for 20 ms. Therefore, in the simulation Kl is delayed from being energized (represented by a "1") for 20 ms. Similarly, should the copper-path open up as a result of a change in any of the variables, K2, K3 or 28V, the relay Kl, will try to de-energize. Mechanical action will delay the contacts from changing by 10 ms. Therefore, in the simulation Kl will be delayed from being de-energized (represented by a "0") for 10 ms.

3/WRITING EQUATIONS

The methods used to write this model from the basic concept of the elementary Boolean equations are explained step by step in the following paragraphs.

3.1 ALL VARIABLES ARE DESCRIBED

Given the following section of a circuit, it is described as follows:

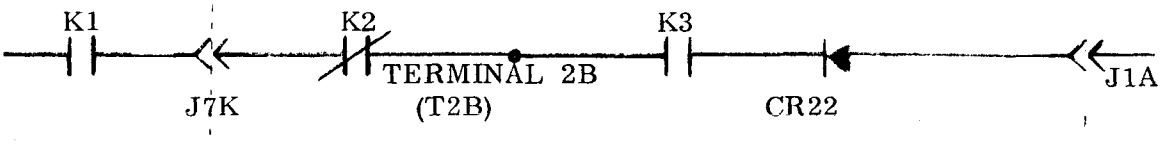


FIGURE 1-4

K1 * /J7K * /K2 * /T2B * K3 * /CR22 * /J1A

Similar to the previous representation of contacts and switches, the "logical negation " (slash symbol) of the extra variables only indicates that those variables are normally closed in their de-energized state (as shown on the drawing). In addition, if these types of variables are sensed in the simulation, because they are "logically negated" no additional logic or commands are necessary for normal or correct functioning of the system. However, the including of all variables in the original writing of the Boolean equations provides three additional benefits:

1. More valid simulation and malfunction analysis of the system.

2. More complete picture of the circuit for automatic display purposes for use in a display system such as the SC-4020.

3. More consistency provided by insuring that all model builders describing the same system include in the model all variables instead of certain specified types.

3.2 ALL VARIABLES ARE WRITTEN IN SEQUENCE

Consider again the previous circuit:

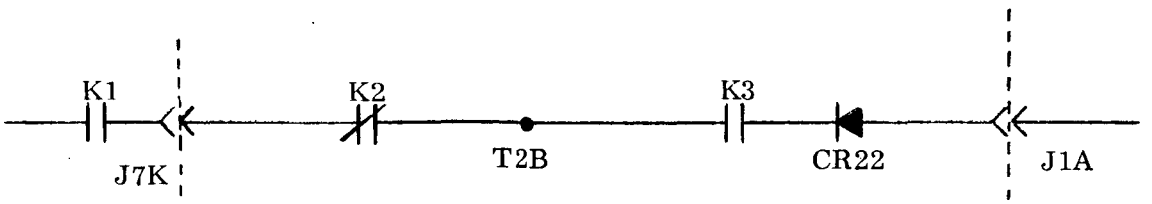


FIGURE 1-5

The Boolean equation for this circuit was:

K1 * /J7K * /K2 * /T2B * K3 * /CR22 * /J1A

Now notice that in going from left to right, the sequence of variables in the equation follows exactly the position of the variables in the circuit.

3.2.1 As will be discussed later in this report, this equation will be written in the opposite direction, such as:

/J1A * /CR22 * K3 * /T2B * /K2 * /J7K * K1

Regardless of which direction the equation is written, it is to be noted that the variables consistently follow each other sequentially as shown in the circuit drawing.

3.2.2 The sequencing of variables within the equation has the following advantages:

1. As a consistency rule, all equations will be written in the same order.

2. If the system is to be pictorially displayed, the display will correspond exactly to the sequencing shown on the circuit drawing.

3.3 LEG-NODE RELATIONS

Given the following circuit:

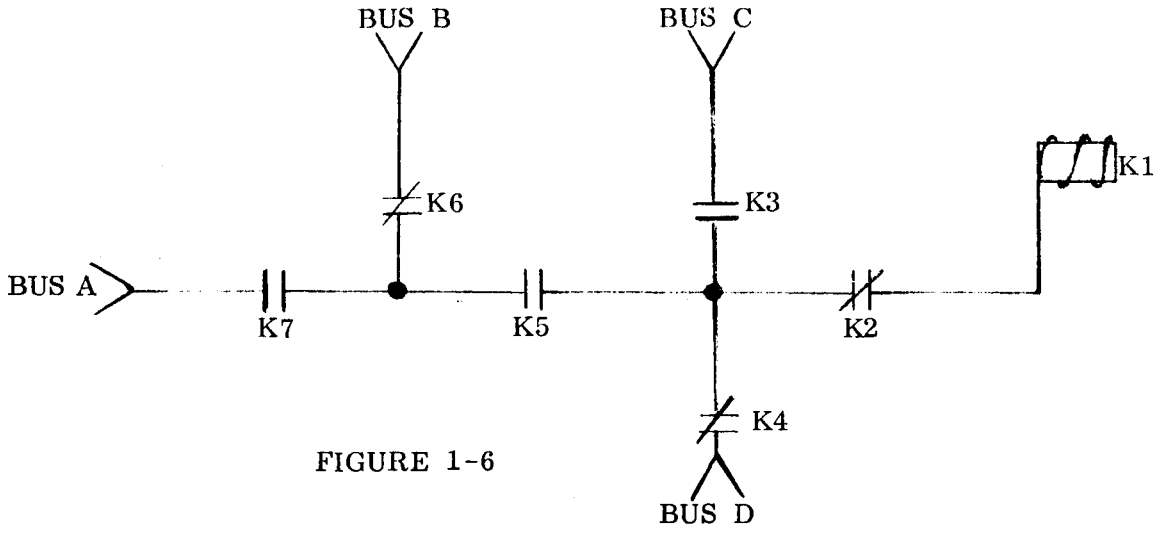


FIGURE 1-6

The basic Boolean equation used to describe this circuit is:

$$K1 = /K2 * (K3 * \text{Bus C} + /K4 * \text{Bus D} + K5 * (/K6 * \text{Bus B} + K7 * \text{Bus A})) \quad (4)$$

All equations are written from the variable being described to the first source of energy such as a "bus." In the above case, since there were four parallel paths, there were four "first" sources of energy, or four buses. This type of equation writing cannot only be cumbersome because of possible lengthy equations with many parallel paths, but also creates redundant writing. For example, if another coil, say Kx, were connected in parallel with K1, the equation for Kx would be an exact copy of the equation for K1, a complete redundancy.

3.3.1 The method used to describe this circuit introduces two additional type variables used throughout the model, namely:

1. The LEG
2. The NODE

They are defined as follows:

1. The LEG -- a single serial path between two nodes, a node and an initiator (source), a node and a terminal (lamp), or an initiator and a terminal.

2. The NODE -- the intersection of three or more legs.

3.3.2 Adding a little more labeling to the last diagram generates the following:

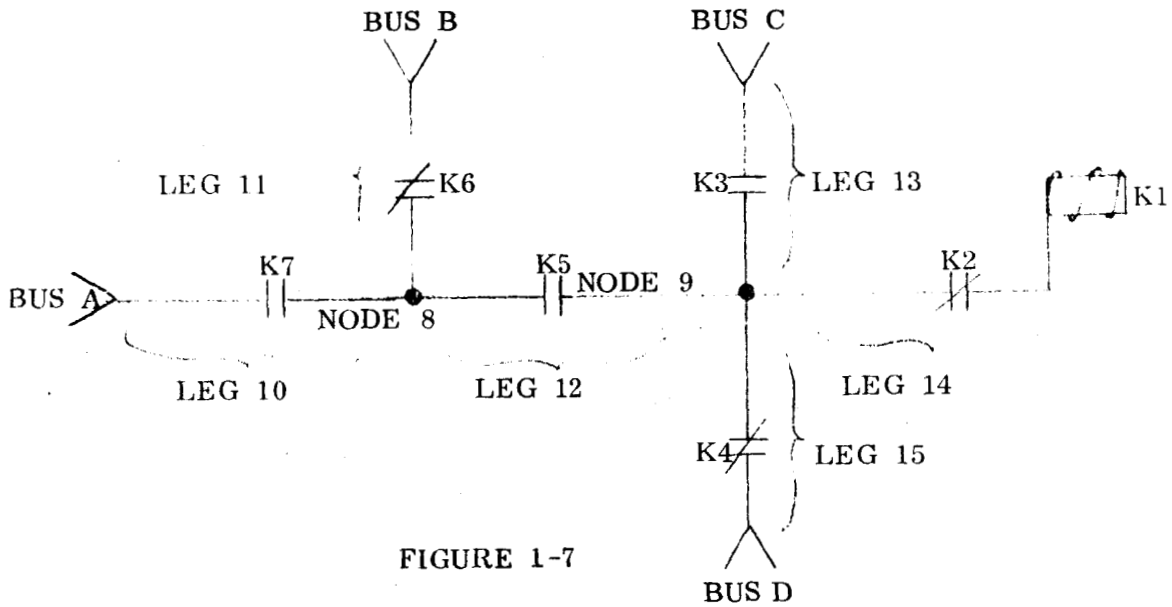


FIGURE 1-7

Figure 1-7 shows that there are only two nodes and six legs. Node 8 has three legs -- 10, 11, and 12 -- while Node 9 has four legs -- 13, 14, 15, and the common leg, 12.

3.3.3 The new equations describing this circuit now are:

Node 8 = Leg 10 + Leg 11 + Common Leg 12 (5)

Node 9 = Leg 13 + Leg 14 $\pm$ Leg 15 + Common Leg 12 (6)

Leg 10 = K7 * Bus A (7)

Leg 11 = /K6 * Bus B (8)

Leg 13 = K3 * Bus C (9)

Leg 14 = /K2 * K1 (10)

Leg 15 = /K4 * Bus D (11)

$$\text{Common Leg 12} = \begin{matrix} \text{K5} * \text{Node 8} \\ \text{or} \\ \text{K5} * \text{Node 9} \end{matrix} \qquad (12)$$

The next section will show how the common Leg 12 is handled. The leg-node concept is similar to the type of equations developed by Kirchoff for circuit analysis. It allows systems or circuits to be described as two-way paths (bilateral).

3.4 BILATERAL TECHNIQUES

Given the following circuit:

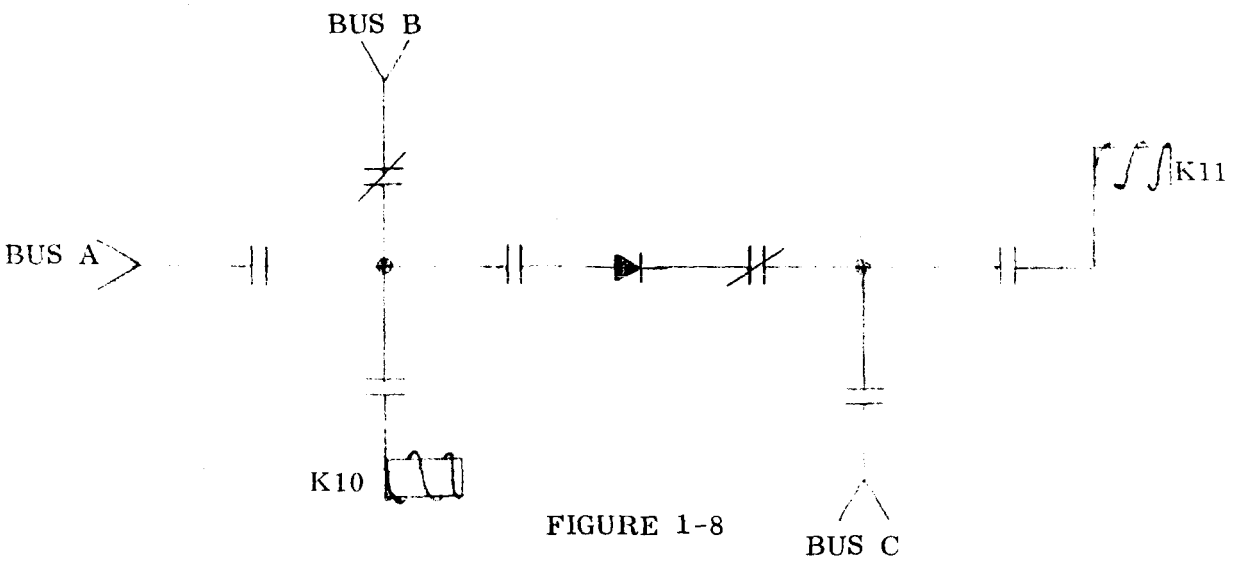


FIGURE 1-8

In the basic method of writing Boolean equations, only an energized electrical path (copper-path) from Bus A or Bus B could energize Coil K 10. This was due to the fact that all equations were written in a "forward manner," i.e., "normal flow of conventional current," or "from higher potential to a lower potential direction," or in a manner expressing only the "normal functioning of the system." Thus the Boolean equations could not show that Coil K10 could be energized by a copper-path from Bus C. But now suppose the diode shorts and a malfunction analysis is required. There is then a need to write many of the equations not only in a "forward-manner" but also in a "reverse manner" to show these "sneak-circuits."

3.4.1 Now suppose the circuit looked like this:

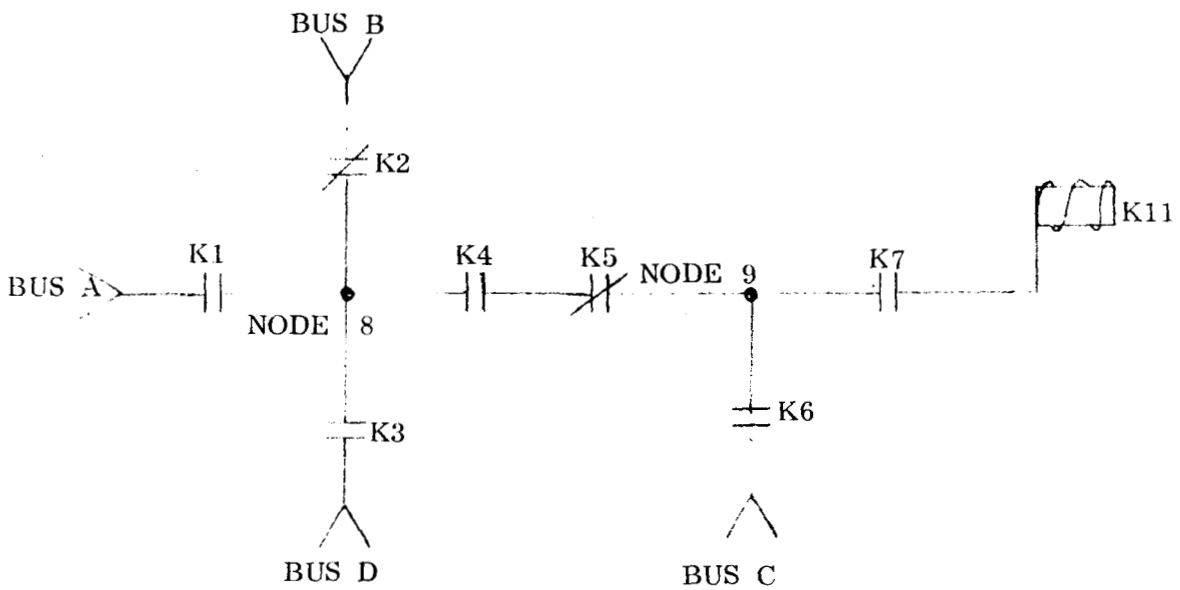


FIGURE 1-9

Even with the diode out of the circuit, with the previous equations, Node 8 could not be energized from the direction of Node 9. But these equations can be written bilaterally, that is, from two sides. From the back to the front (Forward) and from the front to the back (Backward). These equations might then look like:

(Forward): Node 9 = /K5 * K4 * Node 8 (13)

(Backward): Node 8 = K4 * /K5 * Node 9 (14)

But from the previous section it has been shown that:

1. Legs depend upon variables.

2. Nodes depend upon legs.

So these forward and backward equations are really forward and backward legs. They can be expressed as:

Leg of NODE 9 from NODE 8 = /K5 * K4 * NODE 8

Leg of NODE 8 from NODE 9 = K4 * /K5 * NODE 9

3.4.2 To avoid the apparent effect of current flowing in two directions simultaneously and, in addition, the inevitable "hold-in" feature inherent with this set of equations, they should be written as:

Leg (NODE 9 - NODE 8) = /K5 * K4 * NODE 8*/Leg (NODE 8 - NODE 9)

Leg (NODE 8 - NODE 9) = K4 * /K5 * NODE 9*/Leg (NODE 9 - NODE 8)

The node equations should be placed before the leg equations in the model so that the Boolean equations will give the right indications during the simulation when the system is being de-energized. Not all legs have to be described in a bilateral manner. Two types of legs that can be written one way only are:

1. Initiating Legs

   These are legs which are normally energized by an initial variable such as a bus or a source for which there is no further logic to describe this initial variable.

2. Ground Legs

   These are legs which are composed of variables from the grounded side of the hardware. This type of circuitry is further explained in a later section of this report.

3.4.3 It can be seen that legs not only have a name but also a direction associated with them. To minimize the name length of this type of variable it can be decided to:

1. Number all legs around all nodes independently and sequentially starting at "1" for each node at "12 o'clock" and going clockwise.

2. Within the name of these legs, replace the first half or the half telling the direction of the leg by this numbered leg.

For example:

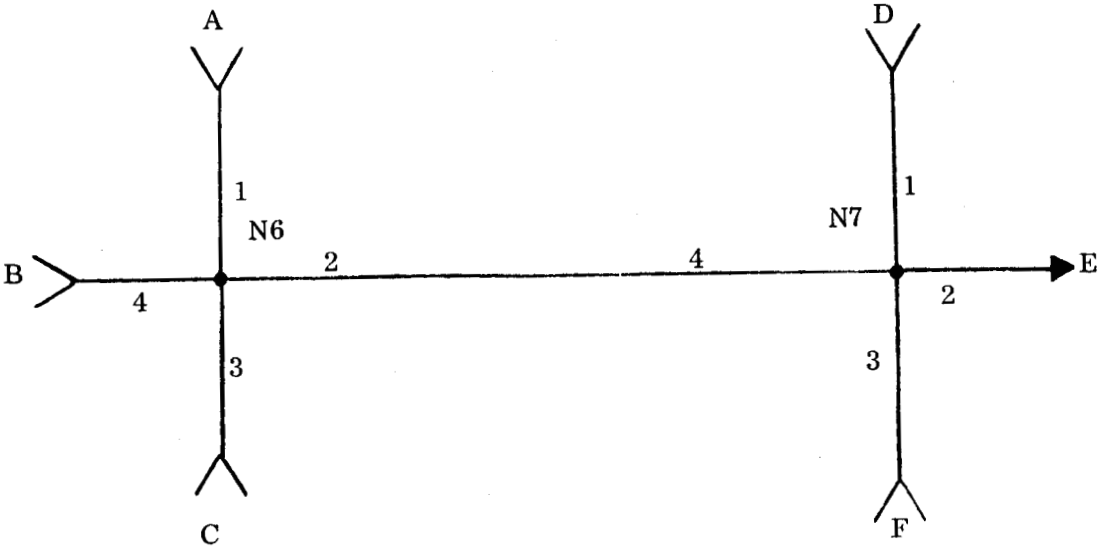


FIGURE 1-10

(Note: There are two numbers for common legs between two nodes.)

The designation of the leg equations of the two nodes shown in Figure 1-10 is as follows:

$$\text{Leg (N6-A} = \text{LEG1N6} \tag{15}$$

$$\text{Leg (N6-N7)} = \text{LEG2N6} \tag{16}$$

$$\text{Leg (N6-C)} = \text{LEG3N6} \tag{17}$$

$$\text{Leg (N6-B)} = \text{LEG4N6} \tag{18}$$

$$\text{Leg (N7-D)} = \text{LEG1N7} \tag{19}$$

$$\text{Leg (N7-E)} = \text{LEG2N7} \tag{20}$$

$$\text{Leg (N7-F)} = \text{LEG3N7} \tag{21}$$

$$\text{Leg (N7-N6)} = \text{LEG4N7} \tag{22}$$

3.4.4 Consider again one of the previous circuits:

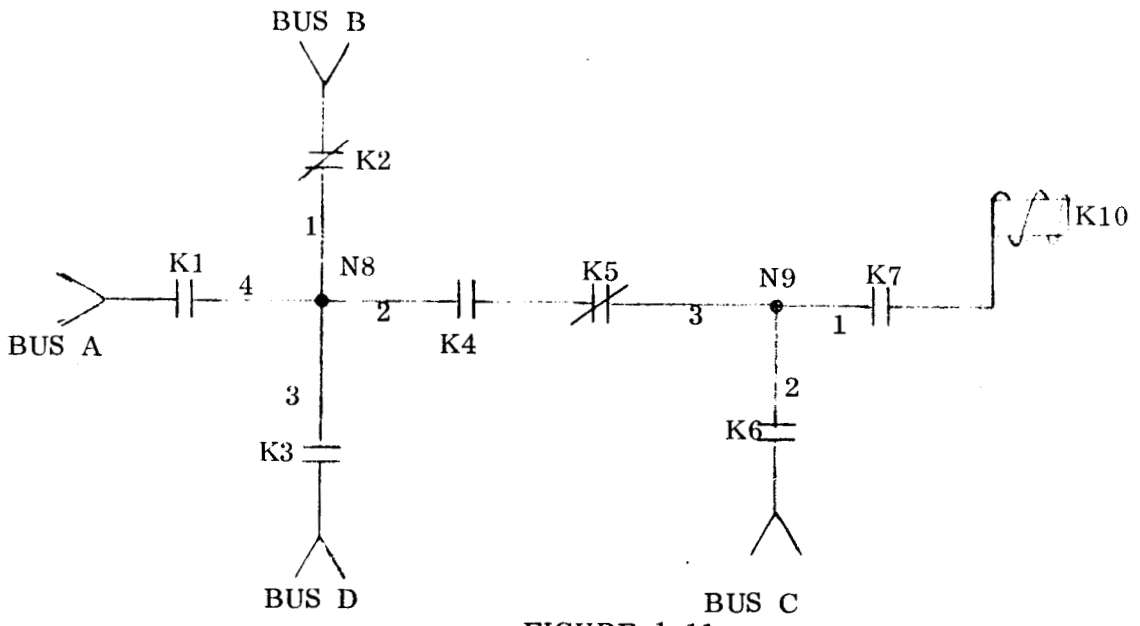


FIGURE 1-11

In summary, the node equations of this circuit are:

$$N8 = LEG1N8 + LEG2N8 + LEG3N8 + LEG4N8 \quad (23)$$

$$N9 = LEG1N9 + LEG2N9 + LEG3N9 \quad (24)$$

The initiating legs are:

$$LEG1N8 = /K2 * Bus\ B \quad (25)$$

$$LEG3N8 = K3 * Bus\ D \quad (26)$$

$$LEG4N8 = K1 * Bus\ A \quad (27)$$

$$LEG2N9 = K6 * Bus\ C \quad (28)$$

The sets of bilateral legs are:

$$LEG3N9 = /K5 * K4 * N8\ */LEG2N8 \quad (29)$$

$$LEG2N8 = K4 * /K5 * N9 * /LEG3N9 \quad (30)$$

and

$$LEG\ (K10\text{-}N9) = K7 * N9 \tag{31}$$

$$LEGIN9 = K7 * K10 \tag{32}$$

The last set of bilateral legs will be further discussed in a later section, since it constitutes a deviation from a simple set of bilateral equations. Notice that not only the variables in the legs are in strict sequence as defined before, but the legs making up the node equations are in a clockwise numerical sequence.

3.5 DIODE ANALYSIS

3.5.1 Previously, diodes were taken into account when writing Boolean equations but they were never considered as a circuit element to be described. Diodes can actually be in any one of three states:

1. Normal
2. Open
3. Shorted

The following circuit can be used as an example:

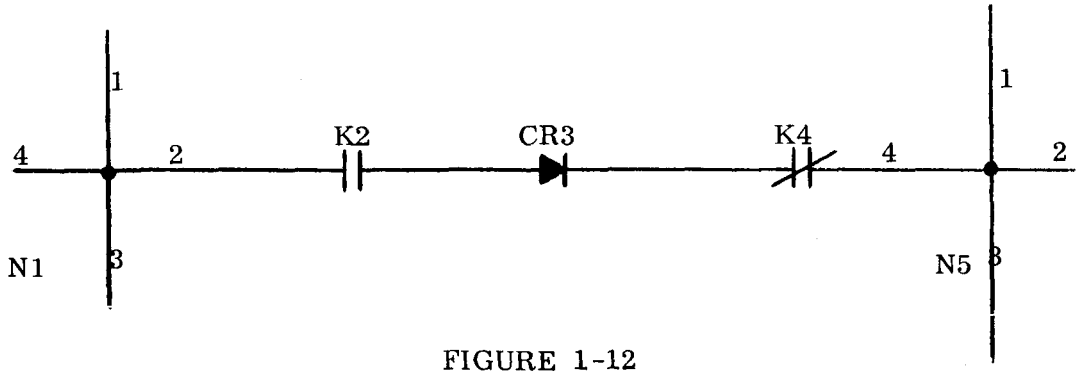


FIGURE 1-12

3.5.2 Since for a diode there are three states (as listed above) instead of two, at

least two variables must be used to describe these three states. By not creating a name for the diode in the normal state and writing the equations for this leg in the following manner, a very efficient set of equations evolves to show the three states of the diode.

LEG4N5 = /K4 * /Diode CR3 Open * K2 * N1 * /LEG2N1 (33)

LEG2N1 = K2 * Diode CR3 Shorted * /K4 * N5 * /LEG4N5 (34)

The results can be tabulated thus:

1. Diode Normal -- no diode inputs needed for simulation. (Only forward leg allowed to energize.)

2. Diode Open -- Diode CR3 Open = 1. (Neither leg allowed to energize.)

3. Diode Shorted -- Diode CR3 Shorted = 1. (Only first leg to energize allowed to energize.)

3.6 COIL-CONTACT RELATIONS

Given the following circuits:

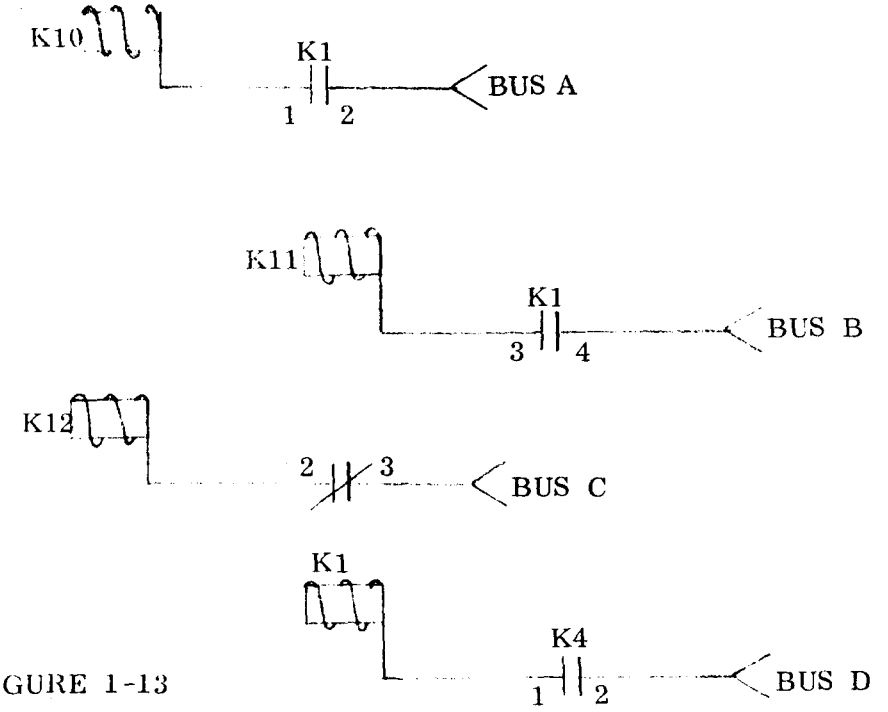


FIGURE 1-13

3.6.1 The coil equations will be written in the following manner:

$$COILK10 = CONTK1C12*BusA \quad (35)$$

$$COILK11 = CONTK1C34*BusB \quad (36)$$

$$COILK12 = 1CONTK1C23*BusC \quad (37)$$

$$COILK1 = CONTK4C12*BusD \quad (38)$$

3.6.2 In addition, the contacts have to be related logically to the coil. This is accomplished by equating each set of contacts of a particular relay to the relay.

$$CONTK1C12 = COILK1 \quad (39)$$

$$CONTK1C34 = COILK1 \quad (40)$$

$$CONTK1C23 = COILK1 \quad (41)$$

3.6.3 When writing those Boolean equations which include contacts as shown on the drawing, it is a good idea to check those contacts with the relay table, showing every set of contacts for each relay and whether the contact set is supposed to be open or closed in the de-energized state. Most drawings will show the contacts and other elements and circuits in the de-energized state.

When writing legs in a bilateral fashion and encountering the contacts of a coil, be sure that the contacts are written in the same order. For example:

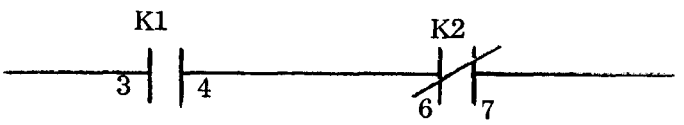


FIGURE 1-14

If the forward equation is:

$$* /CONTK2C67 \quad * CONTK1C34*$$

then the backward equation should be:

* CONTK1C34 * /CONTK2C67*

The pickup and dropout times are placed on all sets of contacts for the coils, while no pickup nor dropout time is placed on the coil.

3.7 GROUND CIRCUITS

Ground circuits should be described as follows:

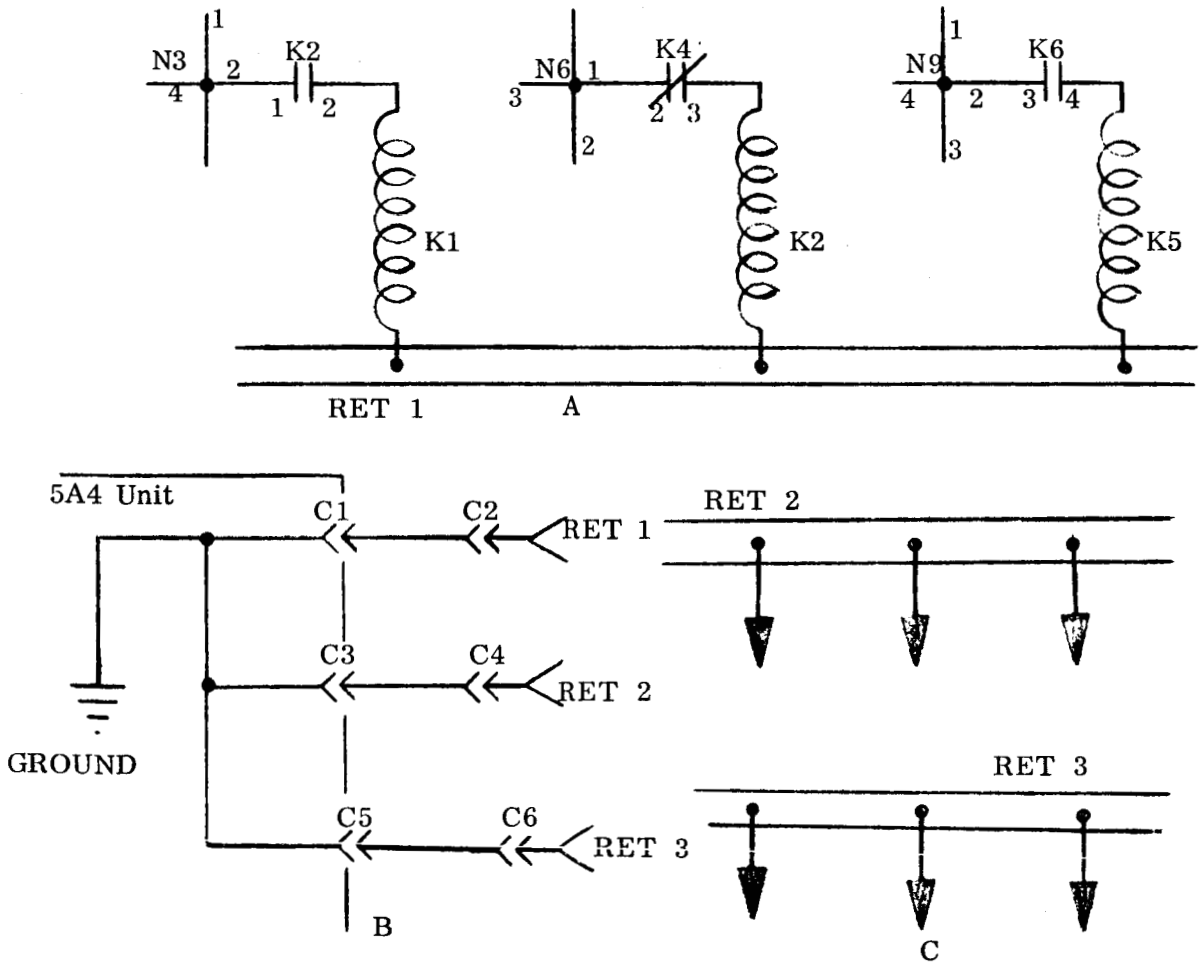


FIGURE 1-15

3.7.1 Previously, ground circuitry has been ignored. It was assumed static, always there when needed. But this is not always true. Sometimes the whole circuit can not be energized until a switch is closed in the grounded side of the circuitry. Also, from a malfunction analysis point of view, there are some types of malfunction that can occur in ground circuitry, and so ground circuitry should be expressed. If ground return is indicated within a particular unit, equations should be written indicating the copper path that leads to the unit.

Previous examples indicated that a variable can be energized any time there is power applied, ignoring the return circuit. To make the model more complete the "Ground" or 'Return' should be included. Reference Figure 1-15.

$$\text{COILK1} = \text{Ret 1} * \text{CONTK2C12} * \text{N3} \tag{42}$$

3.7.2 Ground circuitry illustrated in Figure 1-15 B. need only be described in a unilateral manner eliminating node and leg designations provided that ground connection is in one unit.

$$\text{Ret 1} = /\text{C2} * /\text{C1} * \text{Ground} \tag{43}$$

$$\text{Ret 2} = /\text{C4} * /\text{C3} * \text{Ground} \tag{44}$$

$$\text{Ret 3} = /\text{C6} * /\text{C5} * \text{Ground} \tag{45}$$

In Figure 1-15 the variable "Ground" is an input or initiator, and there is no logical equation describing it within each unit, just as there are no logical equations describing the mechanical connectors, C1, C2, C3, C4, C5 and C6.

3.8 "SINK" EQUATIONS

3.8.1 "Sink" Equations are unilateral equations from a node or bus to power abosorbing variables, such as:

1. Coils
2. Lamps
3. Motors
4. Discrete Inputs
5. Others

3.8.2 A "sink" leg is shown in Figure 1-16.

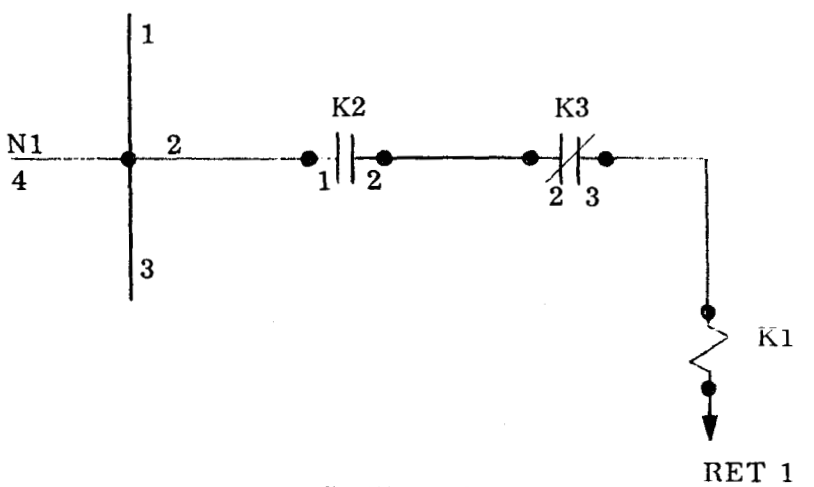


FIGURE 1-16

3.9 BOOKKEEPING CLASSIFICATION OF VARIABLES

All variables can be broken down into the following classifications:

ACTIVE		INACTIVE
DEPENDENT	INDEPENDENT	
Busses	Sources	Terminals
Nodes	Diodes	Permanent Connectors
Legs	Coils	Busses
	Lamps	Sources
	Computer Signals	
	Legs	
Contacts		
Multiple Switches	Single Switches	
Automatic Connectors	Open Semi-Permanent Connectors (Umbil-plug)	Closed Semi-Permanent Connectors (Umbil-plugs)

FIGURE 1-17

Notice that some variables fall into two categories. These will be explained later in this section.

3.9.1 This information on the classification of variables will appear in the last eight columns of each time-card for each variable.

3.9.2 The Active/Inactive division was created to hold to a minimum the number of variables with which the simulation must actively work. When a large system is to be simulated with a great amount of detail, there will be more variables described than the simulation can handle at one time. But many of these variables never change state neither in the preparations for test, nor in the test, nor in post-test. Most of these variables continuously provide a copper-path for the rest of the circuit. If this be the case, these variables are Inactive. All other variables are Active. All the variables can then be classified in this manner and a respective symbol entered

into the last eight columns of a time-card for each variable. The DNS program sorts out all Inactive variables from the model so that the simulation works only with Active variables. Inactive variables are only written in the equation at first for two reasons:

1. This locates them easily in the proper leg (s) and in proper sequence so that if these variables are to be malfunctioned, it can be easily done.

2. If a pictorial display is required, all the variables can be seen.

3. Col. 79 is utilized for "AMA" Failure Candidate Code and is used by that program to determine those variables chosen for failure analysis.

It is interesting to note that if an Inactive variable is to be malfunctioned, it is not necessary to change the classification from Inactive to Active. The leg (s) that uses this variable can be commanded to assume a binary value of zero, and this has the same effect as that element being malfunctioned.

3.9.3 Permanent connectors are those for which the connections are not designed to be opened and closed by hand. Such connectors, when connected with tools, may be sealed to guarantee breaking and making of the connections. On the other hand, semi-permanent connectors have to be manually moved to break and make the connections. However, this can be done quite often with, say, umbilical plugs in the setting up and tearing down of particular tests.

3.9.4 Most dummies are active because they are inserted to give the model a malfunction capability, not easily describable using the variables alone. However, those dummies assigned to an unused leg are inactive. An example follows:

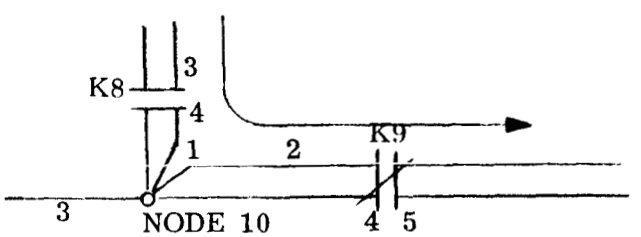


FIGURE 1-18

The double-line path and arrow indicate the copper-path actively used in the model. To show that Node 10 is really a node and not just a terminal, the unused path (LEG3N10) is named in the node equation for Node 10. Because it never changes state, it can be classified as a dummy variable in the Inactive list. Should this path ever be used, the logical node equation is correct. A logical equation for LEG3NODE 10 has to be generated and the classification of this variable must be changed.

Another deciding point for variables -- like grounds, returns, buses, and sources -- is if there is a logical equation for these variables and in this equation leg there is an active variable, then this variable in question is active; otherwise, it is inactive.

3.9.5 The Dependent/Independent division was created to supply an additional classification of all active variables. For the automatic malfunction analysis program this classification will either provide information required to do the job or check the results.

Legs are Dependent if they have dependent variables within them; otherwise, they are Independent.

3.9.6 Another division is the Normally Open/Normally Closed classification. This is only asked of such variables as contacts, switches, semi-permanent and automatic connectors or those variables that easily show their open or closed state on drawings. This classification also will aid in the automatic malfunction analysis program. In addition, this classification may be useful for future pictorial display of the system.

3.9.7 A point to consider in the classification of variables is that for a true pictorial display of the system, all variables must be classified as to what they are, not how they are written. For example, a bus many times acts like a node and if in the model a bus does so, it should be written in the format of a node equation. However, this bus should still be classified as a bus.

3.10 BOOKKEEPING CLASSIFICATION OF EQUATIONS

3.10.1 All equations can be broken down into the following classifications:

1. Bilateral Same (BS) - A leg that is logically the same in either direction of current flow. See Leg 2 Node 8, Figure 1-7, Section 3.3.2.

2. Forward Different (FD) - A leg that is logically different in both directions of current flow. This leg, however, describes the forward or normal path. See Figure 1-12 of Section 3.5.1.

3. Reverse Different (RD) - Same leg as FD but logically different and described in a reverse sequence.

4. Forward Unilateral (FU) - A leg describing an electrical copper-path in the forward direction to a power absorbing variable. See Figure 1-16 of Section 3.8.2.

5. Unilateral Mechanical (UM) - A mechanical relation between two sets of contacts or a set of contacts and its coil. See Coil-Contact Relations of Section 3.6.

6. Node (X) - An equation describing a node as a function of legs connected by logical ORs. See "Node 9" of Figure 1-15, Section 3.7.

7. Unilateral Ground (UG) - A leg written unilaterally in the ground side of circuits. See "Ret 1" of Figure 1-15, Section 3.7.

3.11 FORMAT FOR EQUATION-CARDS

The format for the equation cards is as follows: The variable name will occupy the columns from Col. 1 through Col. 24, with the equal sign occupying any column from Col. 7 through Col. 25. The equation will occupy any or all columns from the equal sign through Col. 72. Columns 73 through 80 will be reserved for bookkeeping information.

3.12 FORMAT FOR TIME-CARDS

The variable name will occupy columns as shown below:

Variable Name (cols. 1-24)	Blank (Cols) 25-36	Pick-Up Times (cols. 37-54)			Drop-Out Times (cols. 55-72)			Bookkeeping (cols. 73-80)
		MIN	AVE	MAX	MIN	AVE	MAX	
		6 Cols.	6 Cols.	6 Cols.	6 Cols.	6 Cols.	6 Cols.	

FIGURE 1-19

3.13 BOOKKEEPING LAYOUT FOR DNS PROGRAM

3.13.1 Bookkeeping layouts used in the DNS program are as follows:

1. Col. 73 - A = Active and I = Inactive
2. Col. 74 - D = Dependent and I = Independent
3. Cols. 75 & 76 = Type of equation, as determined from the classification guide on Page 26 of this report.
4. Col. 77 = Not used.
5. Col. 78 = P used when coil has diode in parallel.
6. Col. 79 = Type variable, as coded from Pages 29 and 31 of this report.
7. Col. 80 = 0 if element is normally open or de-energized and 1 if element is normally closed or energized.
8. The page numbers of the drawing from which the equation started are shown in columns 73, 74, 75, and 76 of the equation card.
9. The sequence numbers of the equation written from the drawing shown in columns 73, 74, 75 are shown in columns 77 and 78 of the equation card.
10. The cards required for equations are numbered in sequence and are shown in columns 79 and 80 of the equation cards.

Code Designation for Cols. 73-80 of Time Cards

Variable	Active Inactive Col. 73	Dependent Independent Col. 74	Type of Equation Cols. 75 76	Coil with Diode Col. 78	Type Variable Code Col. 79	Normal State of Variable Col. 80
Accelerometer			I		A	1/0
Bus	A	I	FU/I		B	1/0
Contact	A	D	U M		C	0
Diode Open	A	I	I		D	0
Dee	A/I	I	F U		E	0
Flite Comb Mon	A	I	F U		F	0
Fuse	A/I	I	I		G	0
Heater	A	D	U M		H	0
Transducer	A	I	I		I	1/0
Discrete In	A	I	F U		J	0
Coil	A	I	F U	P	K	0
Lite	A	I	F U		L	0
Meter	A/I	I	F U		M	0
Solenoid	A	I	F U		N	0
Discrete Out	A	I	I		O	0
Pin	I	I			P	0
Plug	I	I			Q	0
Motor	A	I	F U		R	0
Switch	A	D/I	FU/I		S	0
Timer	A	I	F U		T	0
Dummy	A	I	I		U	0
Valve	A	I	F U		V	0
Test Point	A	I	F U		W	0
Node	A	D	X		X	0
Leg	A	D/I	See Note 1		Y	0
Sensor	A	I	FU/I		Z	0
Input	A	D/I	I		0	1/0
Diode Shorted	A/I	I	I		6	0
Umbilical Pin	A/I	I	F U		7	0
Telemetry Chan.	A	I			8	0
Resistor	I	I	I		9	0

Note 1

Leg Coding in Col. 75 - 76

Bilateral Same	B	S
Forward Different	F	D
Reverse Different	R	D
Forward Unilateral	F	U
Unilateral Mechanical	U	M
Unilateral Ground	U	G
Unilateral Source	U	S
Input		I

Code Designation For Column 79 of Time-Cards
AMA Failure Candidate Codes

	Col. 79
Accelerometer	A
Bus	B
Contact	C
Diode Open	D
Dee	E
Flt. Comb. Monitor	F
Fuse	G
Heater	H
Discrete In	I
Transducer	J
Relay Coil	K
Light	L
Meter	M
Solenoid	N
Discrete Out	O
Pin	P
Plug	Q
Motor	R
Switch	S
Timer	T
Dummy	U
Valve	V
Test Point	W
Node	X
Leg	Y
Sensor	Z
	1
	2
	3
	4
	5
Diode Shorted	6
Umbilical Pin	7
Telemetry Channel	8
Resistor	9
Input	0

TABLE 3

3.14 CONSISTENCY IN BUILDING THE MODEL

Consistency is of greatest importance in building a model. Variables must be designated by the exact letters and digits in each place in which they appear, both in equation and time parameters. Legs of nodes should be numbered in the same manner by all persons writing a particular model, so that interface equations are written the same.